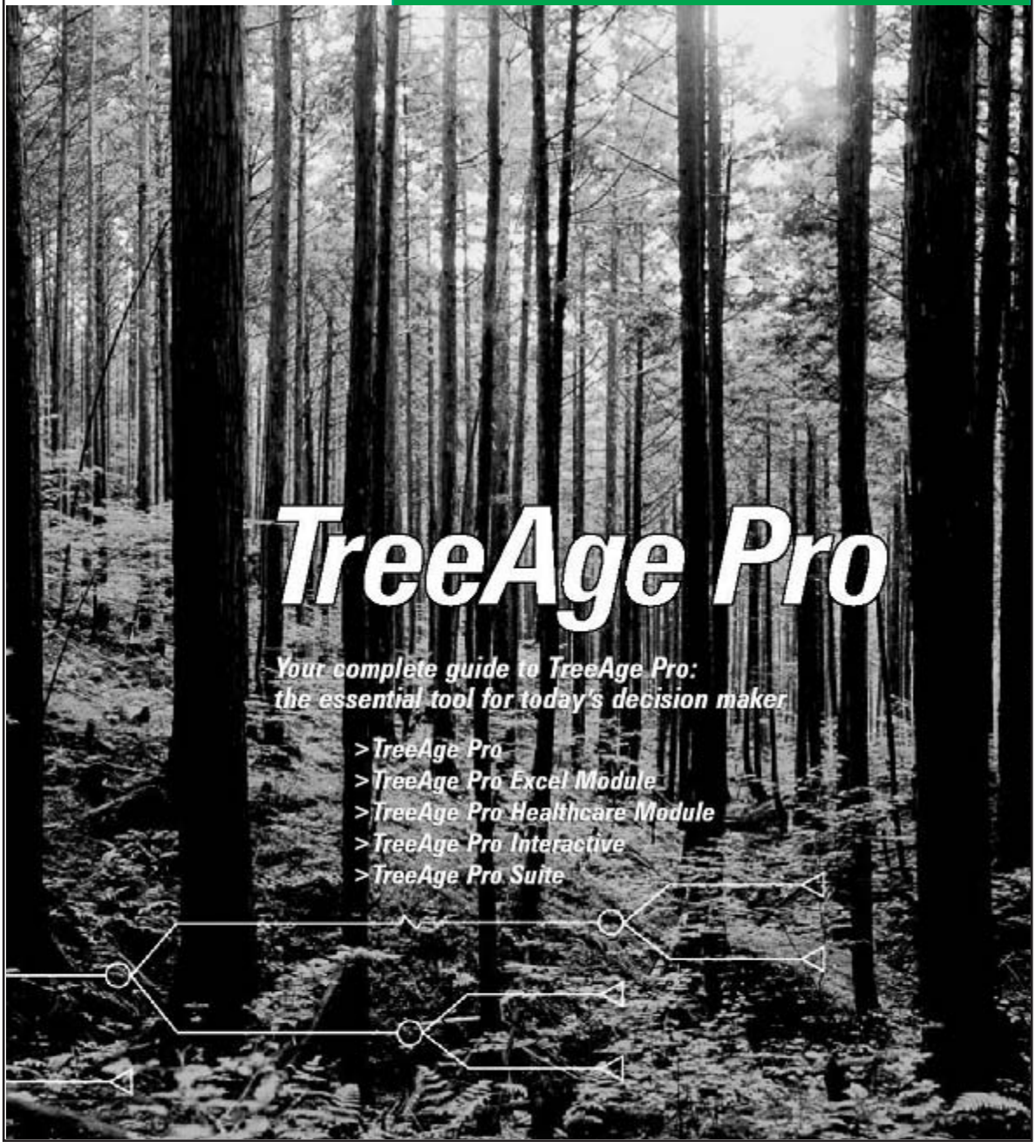
TREEAGE SOFTWARE, INC.

TreeAge Pro 2007 User's Manual



Copyright © 1988-2007 by TreeAge Software, Inc. All rights reserved. No part of this manual may be reproduced in any manner or translated into another language without the express, written permission of TreeAge Software, Inc.

TreeAge Pro, TreeAge Pro Interactive, Decision Analysis by TreeAge, DATA, DATA Professional, DATA Pro, and DATA Interactive are trademarks of TreeAge Software, Inc. Other product names mentioned herein may be trademarks and/or registered trademarks of the respective holders.

The computer program ("Software") described in this user's manual is owned by TreeAge Software, Inc. The Software may be used or copied only in accordance with the terms of a License Agreement which accompanies the Software. Anyone installing, copying, or otherwise using the Software is bound by the terms of that License Agreement.

The License Agreement sets forth a warranty which is expressly limited to the media on which the Software is provided and also contains a limitation on liability. Please do not use the Software in any way without first reading, and concluding that you are willing to be bound by the terms of, the License Agreement.

Nothing in this user's manual, which is informational only, is intended to expand the limited warranty specified in the License Agreement or to act as a representation or warranty of merchantability or fitness for a particular purpose.



1075 Main Street
Williamstown, MA 01267

Voice: 413.458-0104
Fax: 413.458-0105
E-mail: info@treeage.com
Web: http://www.treeage.com

Contents

INTRODUCTION 1

Connecting TreeAge Pro with other software 1

Optional modules 1

Installation and system requirements 1

Getting technical support 2

- Using the PDF manual 2
- Web site resources 2
- Example models 2
- E-mail and telephone support 2

File compatibility 3

Organization of this manual 3

PART I: GETTING STARTED WITH TREEAGE PRO 2007 5

CHAPTER 1: DECISION ANALYSIS PRIMER 7

What is decision analysis? 8

- Modeling 8
- Expected values and optimal decisions 8
- Expected utilities and certainty equivalents 9
- Sensitivity analysis and Monte Carlo simulation 9

A simple problem: How should I invest $1,000? 10

Decision trees 10

- Tree structure 11
- Probabilities and payoffs 11
- Calculating the tree 12

Influence diagrams 13

- Influence diagram structure 13
- Analyzing influence diagrams 14

Further reading 14

CHAPTER 2: A DECISION TREE TUTORIAL 15

The tree window 16

Constructing a tree 17

- Selecting and deselecting a node 17
- Entering node text 18

Adding branches (and nodes) 18
Deleting branches 18
Navigating the tree using the keyboard 19
Saving the tree 20
Entering payoff values 20
Entering probabilities 22
Regional (e.g., European) numeric settings: 22
Setting calculation and numeric formatting preferences 23
Calculating the tree 24
Roll back text report 24

CHAPTER 3: AN INFLUENCE DIAGRAM TUTORIAL 27
Constructing an influence diagram 28
The influence diagram window 28
Creating and selecting nodes 29
Entering node text 29
Adding arcs 30
Arc shortcuts 31
Editing existing nodes 31
Selecting and editing existing arcs 31
Entering hidden information 32
Assigning alternatives and outcomes 32
Entering probabilities 34
Entering values 35
Creating asymmetry: arc information 36
Converting and calculating the model 38
Notes on conversion 39

CHAPTER 4: PRINTING AND PRESENTING TREES AND INFLUENCE DIAGRAMS 41
Printing 42
Adjusting printouts using print preview 42
Headers and footers 43
Printing preferences 43
Global preferences 43
Exporting pictures 44
Copying a picture of an influence diagram or graph 44
Copying a picture of an tree 45
Exporting a picture to a file 45
Working with exported graphics in other programs 46
Roll Back (.TRB) files 47

PART II: WORKING WITH DECISION TREES 49

CHAPTER 5: ANALYZING DECISION TREES 51
Numeric formatting 52
Notes on entering numbers 52
Expected values 53
Roll back 54
Customizing the roll back display 54
Rankings 55
Standard deviation 56
Probability distributions 57
Graph window contents 57
Downstream decision nodes 59
Cumulative probability distributions 60
Comparative probability distributions 61
Dominance in probability distributions 63
Expected value of perfect information 65
How EVPI is calculated 65
Avoiding EVPI errors 68
Other analyses 69
Range of possible payoffs 69
Verify probabilities 69
Over/under probabilities 70

CHAPTER 6: GRAPH WINDOWS 71
Customizing graphs 72
Making changes to labels 74
Creating custom labels 75
Making changes to numeric axes options 76
Adding a horizontal or vertical line 78
Viewing and exporting graph data 79
Viewing a graph's text report 79
Printing and exporting text reports 80
Details on specific graph types 81
Bar graph 81
Line graphs (and cost-effectiveness graphs) 82
Scatterplots 84
Tornado diagrams 84
Region graphs 84
Using graph templates 85
Creating and applying graph templates 85

CHAPTER 7: TREE CALCULATION METHODS AND PREFERENCES 87
Changing what the tree calculates 88
Selecting calculation method 89
Optimal path 90
Calculate extra payoffs 91
Numeric formatting 92
Calculations using multiple attributes 93
How benefit-cost calculations are performed 94
How weighted multi-attribute calculations are performed 94
Roll back analysis options 95
Maximin (and Minimax) roll back 96
Regional/international numeric settings (decimals and thousands separators) 97
Overriding the optimal path at selected nodes 98
Force path 98
Change optimal path 98

CHAPTER 8: SELECTING SUBTREES AND MULTIPLE NODES 99
Selecting a subtree 100
Selecting multiple, unrelated nodes 100
Selecting multiple nodes by characteristic 101

CHAPTER 9: MAKING CHANGES TO TREE STRUCTURE 103
Tree-building commands – a review 104
Inserting, deleting, and reordering branches 105
Cut, copy, paste, and clear endnodes and subtrees 108
Using the tree clipboards 110
Cut, copy, paste, and clear text 111
Undo and Redo 112
Reverting to a file's saved state 112
Find and replace text, formulas, and values 113
Using the Probability Wheel 114

CHAPTER 10: ANNOTATING THE TREE 115
Label nodes 116
Hidden node comments 117
Editing comments in the Properties grid 117
Note boxes and arrows 118
Creating note boxes 118
Binding a note box to a node 119
Creating arrows 120
Binding arrows to nodes 120

CHAPTER 11: TREE DISPLAY PREFERENCES AND OPTIONS 121
Collapsing/hiding subtrees 122
Expanding hidden subtrees 122
Aligning selected nodes 123
Displaying roll back columns 124
Copying columns to Excel 126
Custom roll back calculations 127
Other tree display preferences 128
Hiding and boxing payoffs 128
Hiding probabilities and branch labels 128
Displaying a "skeleton" tree 128
Terminal node numbers 129
Number all nodes 129
Increasing or decreasing vertical white space 130
Vertically aligning terminal nodes 131
Changing fonts 132

PART III: UNCERTAINTY AND VARIABILITY:
USING VARIABLES, TABLES, AND DISTRIBUTIONS 135

CHAPTER 12: INTRODUCTION TO VARIABLES AND SENSITIVITY ANALYSIS 137
Sensitivity analysis background 138
Variables and sensitivity analysis 138
Using variables in a tree 139
Steps for using variables 139
Guidelines for naming variables 139
Creating and defining variables 140
Finding and fixing problems with definitions 143
Adjusting complementary probabilities 144
Performing one-way sensitivity analysis 148
Setting the sensitivity analysis range 148
The sensitivity analysis graph and report 150
Sensitivity analysis thresholds 151
The Threshold Values legend 151

CHAPTER 13: WORKING WITH VARIABLES 153
Variable Categories 154
Variables Report 157
Variables reporting options and calculations 157
Variables testing tools 158
Debugging variable calculations during analysis 158
The Calculator/Evaluator 158
Variable sliders 159

Formula Editor and Insert Variable dialogs 160
Variable Definitions window 161
Updating and adding variable definitions 162
Changing the node selection 162
Using the Show inherited option 162
Cut, copy, and paste variable definitions 163
Variables and Tables dialog 164
Selecting multiple variables 165
Importing a list of variables from another tree 165
Edit in Excel (Excel module) 165
Variable Properties dialog 166
Sensitivity analysis correlations 167
The Define Variable window 168
Using the Functions Helper 169
User-defined Python functions (new in v2007) 169

CHAPTER 14: BUILDING FORMULAS USING VARIABLES AND FUNCTIONS 171
Variable formulas 172
Building a complex cost formula: an example 172
Implementing a cost formula using variables 173
Variables with multiple definitions 174
Using node-specific variable definitions 175
Using mathematical, statistical, and other functions 176
Notes on using functions in formulas 176
Arithmetic functions 177
Financial/discounting functions 178
Miscellaneous functions 179
Recursive variable definitions 180

CHAPTER 15: MORE SENSITIVITY ANALYSIS TOOLS 185
Analyzing variables with multiple definitions 186
Tornado diagrams 188
Incremental calculations 190
Additional calculations in the text report 190
Two-way sensitivity analysis 191
Choosing intervals 192
Isocontours 193
Three-way sensitivity analysis 194
Threshold analysis 195
How threshold analysis works 195
Threshold tolerance 196
Threshold analysis non-linearity 196

Analyzing correlated variables 198
Additional sensitivity analysis topics 199
Sensitivity analysis on variables with non-numeric definitions 199
Checking probability coherence 200
Analyzing a single option 200

CHAPTER 16: USING DISTRIBUTIONS AND MONTE CARLO SIMULATION 201
Uses of Monte Carlo simulation 202
Probabilistic sensitivity analysis (sampling) 202
Microsimulation (first-order trials) 203
Nonlinearity 204
EVPI/value of information analysis 204
Creating distributions 205
Defining a parameter using a distribution 205
Managing distributions 208
Performing probabilistic sensitivity analysis 209
Saving Monte Carlo simulations 210
Simulation reports and graphs: the basics 211
Expected Value of Perfect Information (EVPI) 212
Simulation options 214
Distributed simulations 214
Multi-threading for faster simulations 214
Conserving memory 214
"Seeding" the random number generator 215
Turning off sampling of selected distributions 215
Identifying simulations 216
"Downstream" decision nodes during simulations 216
Recalculate using first-order trials (microsimulation) 217
Three-dimensional simulations 217

CHAPTER 17: DISTRIBUTION FUNCTIONS, OPTIONS, AND TYPES 219
Distribution functions 220
Distribution options 221
Changing sampling rate for microsimulation models 221
Sampling during non-simulation, EV calculations 222
Defining distribution parameters non-numerically 222
Approximating distribution parameters from statistical information 223
Commonly-used distributions 224
Normal distribution 224
Uniform distribution (real number form) 224
Uniform distribution (integer form) 224
Triangular distribution 224
Fractile distributions (10/50/90 et al) 225

More continuous distributions 225
Beta distribution (integer form) 225
Beta distribution (real number form) 225
Dirichlet distribution (multivariate, normalized Beta) 225
Chi distribution 226
Chi-Squared distribution 226
Erlang distribution 226
Exponential distribution 226
Gamma distribution 227
Hyper-exponential distribution 227
Laplace distribution 227
Logistic distribution 227
Lognormal distribution 228
Maxwell distribution 228
Rayleigh distribution 228
Weibull distribution 228
More discrete distributions 229
Binomial distribution 229
Poisson distribution 229
Sampling from tables during Monte Carlo simulation 230
Creating a Table-type distribution 230
Using a distribution to lookup values in a table 231
Converting a distribution to a discrete table 232

CHAPTER 18: CREATING AND USING TABLES 233
Creating and editing tables 234
Creating an empty table file 234
Entering values in a table 235
Table window/editing tips: 237
Linking a table to an ODBC data source 238
Using ODBC data sources in Windows 238
Setting a table's data source properties 239
When to requery data sources 240
Referencing tables in formulas 241
Table syntax 241
Table lookup methods 241
Packaged tables versus table files 242
Managing the Tables directory 243
The Tables Used report 244

PART IV: MORE TREE-BUILDING TOOLS AND OPTIONS 245

CHAPTER 19: USING STORED ANALYSES AND ANALYSIS SEQUENCES 247

Using stored analyses 248

Using graph templates with stored analyses 250

Sequencing stored analyses 251

Creating an analysis sequence 251

Editing stored sequence 254

Setting root variable definitions 255

CHAPTER 20: LINKING WITH EXCEL AND OTHER APPLICATIONS 257

Dynamic linking with an Excel spreadsheet 258

Calculating payoffs using dynamic links: an example 258

Naming a spreadsheet cell 259

Linking to a spreadsheet output cell 259

Connecting tree variables to input cells (optional) 261

Using the cell output in the tree: the Bilink() function 262

Calculation using dynamic links 262

Using a single link to a range of cells 262

Using Bilink() to create non-dynamic links 263

Using DDE links 264

DDE basics 264

Using TreeAge Pro as a DDE server 269

CHAPTER 21: TOOLS FOR LARGE AND COMPLEX TREES 271

Working with very large trees 272

Review 272

Additional tools/techniques in this chapter 272

The Explorer view and Properties grid 272

Docking the Explorer view and other panes 272

Explorer view and Properties grid details 274

Tree outlines and the Node Outline window 276

The Node Outline window 277

Notes on using the Node Outline window 277

Tree outline basics 278

Node outline details 278

Branch name and branch probability 278

Node types 279

Payoffs 279

Definitions of variables 279

Cloned subtrees 280

Note boxes 280

Tracker variables 280

Table lookups 280
Markov transitions and transition bindings 280
Markov termination condition 281
Markov state and transition rewards 281
Markov tunnel states 281
Cloning subtrees 282
Creating clone masters and attaching copies 282
Hiding clone copies 286
Optimizing cloned trees 287
User-defined Python functions/variables 288
User-defined function syntax 288
Notes on TreeAge Pro's Python interpreter 289
Node(), Tree(), and other special functions 290
String-arguments and concatenation 290
Using the Node() function 290
Node() function syntax 291
Using the Tree() function 292
The User() and Python functions 293
Command() function 294
Script() function 294
Using the Debug() function and the Debug pane 295
Global() function 296
List/matrix functions 297
BilinkName() function 297
DDE linking trees 298
When not to use DDE linked/nested trees 298

CHAPTER 22: ADVANCED CHANCE NODE TECHNIQUES AND OPTIONS 299
Using "non-coherent" probabilities: 300
Reporting future net present values in the tree 300
Discrete and dynamically-sized Markov cohorts 301
Sampling probabilities from a multivariate Dirichlet distribution 302
Using DistKids() to create an array of subtrees 304
Distributing a fixed set of chance node branches 306

CHAPTER 23: BAYES' REVISION IN DECISION TREES 309
An introduction to Bayes' revision 310
Probability revision using Bayes' theorem 310
Performing Bayes' revision in the tree 312
Entering the probabilities 313
Making changes to the test subtree 317
Valid prior and likelihood probability expressions 317
Node-specific definitions of test probabilities: 317

CHAPTER 24: UTILITY FUNCTIONS AND RISK PREFERENCE 319
Risk preference: an illustration 320
Certainty equivalents and risk aversion 321
Risk preference curves 322
Creating a risk preference function 323
Constant risk aversion 323
Non-constant risk aversion 324
Risk preference notes: 325

PART V: WORKING WITH INFLUENCE DIAGRAMS 327

CHAPTER 25: BUILDING INFLUENCE DIAGRAMS – TIPS AND TOOLS 329
When to use influence diagrams 330
Bayes' revision 330
EVPI (expected value of perfect information) 330
Model size and other considerations 330
Limitations of an influence diagram 330
TreeAge Pro's node conversion algorithm 331
No-forgetting arcs 332
Using graphical position to resolve time order 332
Asymmetry 333
Deterministic nodes, value nodes, and node variables 334
Deterministic nodes 334
Value nodes 335
Node variables 336
Node variables and asymmetry 336
Using the mini-tree assessment window 337
Entering a new variable in the mini-tree editor 337
Using existing variables in the mini-tree editor 338
The mini-tree probability wheel 338
Insert/create distributions in the mini-tree 339
Dynamic linking in the mini-tree 339
Miscellaneous 340
Note boxes 340
Node description 340
Arc operations 341
Aligning nodes 342

CHAPTER 26: INTERMEDIATE/ADVANCED INFLUENCE DIAGRAM TOPICS 343
Bayes' revision in influence diagrams 344
Setting up a single forecast 344
Asymmetry inside the Bayesian model 347
Bayes' revision with sequential tests 347

Expected value of perfect information 348
Clones 349
Sub-models 350
Sub-models and clones 350

EXCEL MODULE

PART VI: WORKING WITH THE EXCEL MODULE 353

CHAPTER 27: LINKING SPREADSHEETS AND TREES USING THE EXCEL/COM MODULE 355
Linking a variable to a cell's value 356
Setting up dynamic, bi-directional links 357
"Linking" a spreadsheet to node calculations 361
Writing macros targeting the TreeAgeProLib 361
TreeCalc( ) worksheet function 361
"Behind the function" 362

CHAPTER 28: GRAPHING, REPORTING, AND MODELING USING EXCEL 363
Exporting TreeAge Pro text reports 364
Creating Excel charts 364
Converting existing TreeAge graphs into Excel charts 365
Customizing distribution charts with the Excel add-in 365
Editing variables, tables, and distributions in Excel 367
Exporting lists of variables 367
Editing and creating variables 367
Notes on editing variables in Excel: 368
Editing and creating tables 369
Editing and creating distributions 370

CHAPTER 29: IMPORTING INFLUENCE DIAGRAMS FROM SPREADSHEETS 371
Importing a simple spreadsheet 372
Setting up the spreadsheet 372
Converting the spreadsheet 373
Customizing the influence diagram 376
Cleaning up imported influence diagrams 381

CHAPTER 30: USING THE TREEAGE PRO 2007 COM AUTOMATION LIBRARY 383
What is the "TreeAgeProLib" COM library? 384
Using the TreeAgeProLib library in a macro 385

HEALTHCARE MODULE

PART VII: COST-EFFECTIVENESS ANALYSIS WITH THE HEALTHCARE MODULE 389

CHAPTER 31: BUILDING AND ANALYZING COST-EFFECTIVENESS MODELS 391

Before you begin 392

Preparing a tree for cost-effectiveness calculations 393

The Cost-Effectiveness calculation method 394

Performing cost-effectiveness analysis 398

The CE analysis text report 399

Dominance and incremental cost-effectiveness 400

Extended dominance and ICERs 401

The threshold ICER (or willingness-to-pay, or ceiling ratio) 403

Extended dominance: an additional perspective 404

One-way cost-effectiveness sensitivity analysis 405

CE sensitivity analysis text report 405

CE sensitivity analysis graphs 406

CHAPTER 32: COST-EFFECTIVENESS MODELING AND ANALYSIS OPTIONS 407

Net benefits calculations 408

Performing Net Benefits calculations 408

Multi-attribute weighted costs 409

Multi-attribute costs: an example 409

Notes on using multi-attribute costs in CE trees 412

Markov CE models using multi-attribute costs 412

Inverting effectiveness and incremental effectiveness calculations 413

Inverting incremental effectiveness 413

Inverting expected effectiveness calculations 415

CE roll back optimal path parameters 416

The CE optimal path algorithm 417

Thresholds and CE sensitivity analysis 418

Net Benefits thresholds in a 1-way analysis 418

Creating an incremental cost-effectiveness tornado diagram 420

Using a Net Benefits tornado diagram 420

Finding cost-effective thresholds using the ICER graph 422

2- and 3-way CE sensitivity analysis thresholds 424

Displaying incremental values during roll back 425

CHAPTER 33: COST-EFFECTIVENESS SIMULATION REPORTS AND GRAPHS 427

Basic CE statistics and simulation outputs 428

Simulation text report 429

CE graphs 430

Distribution graphs 430

Scatterplots 431
The Incremental Cost-Effectiveness (ICE) scatterplot 432
ICE density plots: isocontours and mountain graph 434
Acceptability curves and Net Benefits calculations 436
Net Benefits vs. WTP 437
Net benefits probability distributions 439
Optimality graph 440
Cost-effectiveness value of information (EVPI and EVPPI) 441
EVPI vs. WTP Curve 441

PART VIII: MARKOV MODELING, ANALYSIS, AND MICROSIMULATION WITH THE HEALTHCARE MODULE 443

CHAPTER 34: BUILDING AND ANALYZING MARKOV MODELS 445
Markov modeling basics 446
State transition models 446
Graphical representation 446
Calculation basics 447
Non-standard Markov models 447
Building a Markov cycle tree in TreeAge Pro 448
The Markov node 448
Markov states (and initial probabilities) 449
Transition subtrees and absorbing states 453
The _stage counter and the termination condition 457
Analyzing a Markov model 458
Cohort (expected value) analysis 458
Expanded Report 462
A note on microsimulation 465
A note on half-cycle correction 465
Cost-effectiveness Markov models 466
Cost-effectiveness keywords 467
A cost-effectiveness Markov model 468

CHAPTER 35: MARKOV MODELING TOOLS AND TECHNIQUES 469
Keywords, time-dependence, and discounting 470
Cycle zero 470
Using tables of time-dependent transition probabilities – an example 471
Discounting rewards 473
Probability/rate functions 474
Assigning one-time costs and utilities 476
Half-cycle correction 476
Prior costs 477
Transition rewards 478

Cloning Markov models 480
Using Markov transition bindings 480
Cloning an entire Markov process 483
Counting "time in state" using tunnels 484
Temporary states and the _tunnel counter 484
Using a tunnel state – an example 485
Using the _tunnel counter 487
Merging results for temporary states 487
Populating temporary states at cycle 0 488
Special binding names and tunnels 488
Markov decision processes 491
Concept 491
Implementation 492
Reporting 492
Modeling discrete and/or changing size "populations" during cohort analysis 494
Calculations using state probabilities 496
Nesting or linking Markov models 497
Sampling probabilities from a multivariate Dirichlet distribution 499

CHAPTER 36: MARKOV MONTE CARLO MICROSIMULATION 501
Microsimulation and tracker variables 502
Using tracker variables for reporting 502
Using trackers in Markov calculations 504
Creating and defining tracker variables 504
Tracker variable properties 505
Tracker modifications that reference other trackers 506
Tracker modification timing 506
Trackers and expected value calculations 506
Debugging Markov microsimulations 507
Using the Debug() function and the Debug pane 507
Two-dimensional probabilistic sensitivity analysis using microsimulation 508
Resampling distributions during microsimulation trials 511
Sequences and linear sensitivity analysis 513
Other aspects of microsimulation 515
Running Markov microsimulation trials in parallel 515
Using logic nodes in a Markov microsimulation 516

CHAPTER 37: MARKOV TECHNICAL NOTES 517
Changing the storage location for tables 518
Markov cohort EV calculation details 519
Markov microsimulation trial details 520
Tracker modification timing 520

APPENDICES

APPENDIX A: CHANGES FROM EARLIER VERSIONS 523
What's new in TreeAge Pro 2007 524
What's new in the v2007 Core module 524
What's new in the v2007 Excel/COM module 527
What's new in the v2007 Healthcare module 527
Changes in v2006 528
What's new in the v2006 Core module 528
Changes in the v2006 Excel module 530
Changes in the v2006 Healthcare module 530
Changes from DATA to v2004/v2005 531
Core module additions/changes 531
Changes from v2004 to the v2005 Excel module 535
Changes from DATA to the Healthcare module 535

APPENDIX B: TOOL BAR AND MENU REFERENCE 537
The tool bar 538
Navigation button 538
Tree Window only 538
Influence Diagram only 538
The status bar 539
File menu 540
Edit menu 542
Display menu 545
Values menu 547
Options menu 549
Analysis menu 551
Diagram menu (influence diagram window) 553
Graph Actions menu 554
Table Actions menu 555

APPENDIX C: PREFERENCES 557
Using the Preferences dialog 558
Hot keys when opening the Preferences dialog 558
Preferences categories 567

APPENDIX D: FUNCTIONS AND OPERATORS 575
Operators 576
Arithmetic operators 576
Relational operators 576
Logical operators 577
Operator precedence 577
Using mathematical, statistical, and other functions 578

APPENDIX E: CREATING SCRIPTS AND INTERFACES 579
Using the TreeAge Pro Excel/COM module library 580
Developing TreeAge Pro Interactive™ applications 581
Technical information 582
Using Legacy TreeAgeScript/DATAScript 585
Syntax 585
Script command reference 586
Creating run-time custom interfaces 587
Creating a Basic Custom Interface 589
Testing the Basic Custom Interface 592
Creating an Extended Custom Interface 593
Protecting your intellectual property 595

APPENDIX F: TECHNICAL NOTES 597
Changing the storage location for tables 598
Using TreeAge Pro's command line 599
Using custom start-up flags 599
Startup files 599
Defining variables 599
Influence diagrams 599
Other application options/details 600
Beginner visual cues 600
Suppressable dialogs 600
Window sizing 600
Hot keys when opening a tree 600
Hot keys when opening the Preferences dialog 600
Importing SMLTREE and DecisionMaker trees 601
Import options 601

INDEX 603

Welcome to TreeAge Pro™ 2007

TreeAge Pro 2007 has been designed to implement the techniques of *decision analysis* in an intuitive and easy-to-use manner. It transforms decision analysis from a potentially tedious exercise into an easily applied and highly visual means of (1) *organizing* the decision making process, (2) *analyzing* the problem at hand, and (3) *communicating* both the structure of the problem and the basis for the decision reached.

If you are experienced in decision analysis, you will find TreeAge Pro 2007 easy to use following only a cursory review of the software commands, although the richness of the program will become more apparent with further study of the manual.

If you have no, or only limited, experience with decision analysis, TreeAge Pro 2007 will make it much easier to learn. See Chapter 1 for a basic decision analysis primer. (The organization of the rest of the manual is described at the end of the Introduction.)

Optional modules

TreeAge Pro 2007 follows a modular design. The "Core" module includes both decision tree and influence diagram functionality, Monte Carlo simulation, multi-way sensitivity analysis, and much more. Additional features may be added in the form of two optional add-ons: a Healthcare module and an Excel™ add-in module.

The ***TreeAge Pro Healthcare*** module integrates and updates the healthcare-related features found in our DATA software, including:

- Markov cohort analysis
- Markov Monte Carlo microsimulation
- Cost-Effectiveness analysis
- Net Benefits analysis

The ***TreeAge Pro Excel/COM*** module can help:

- automate the creation of Excel charts and reports from within TreeAge Pro
- simplify the integration of spreadsheets and TreeAge Pro models via dynamic linking and/or COM automation (i.e., macros)
- convert a spreadsheet model into a fully-linked TreeAge Pro influence diagram

The ***TreeAge Pro "Suite"*** includes both the Healthcare and Excel/COM modules.

Integration with other software

In TreeAge Pro 2007, users of the Excel/COM module gain access to powerful features that allow automation of TreeAge Pro tasks via scripts and macros, in order to control updating and analysis of your models. See Chapter 30.

In addition, the embedded Python script interpreter has been improved, giving modelers easy access to the powerful Python programming language. See Chapter 21.

Installation and system requirements

To install TreeAge Pro 2007, please follow the installation instructions available on the installation CD and from our web site, at:

http://www.treeage.com/treeagepro/

The installation instructions also cover hardware and software (operating system) requirements. See the next section for information on how to contact TreeAge Software if you encounter difficulties during installation.

Getting technical support

There are several ways to get help in using TreeAge Pro 2007.

Using the PDF manual

If the PDF manual has been installed, links to it can be found both under TreeAge Pro's Help menu, as well as in the TreeAge Pro program folder under the Windows Start menu.

There are a number of ways to search the Acrobat™ PDF document. The Bookmark list displayed on the left side of the Acrobat window duplicates the manual's Table of Contents; both includes hyperlinks to specific pages. Page numbers in the document's Index are also hyperlinked (to find the Index, scroll to the bottom of the Bookmark list). In addition, the Acrobat tool bar includes a Find button (the binoculars icon), which can locate any text phrase.

The latest version of the PDF manual is available as a free download, in Adobe® Acrobat™ PDF format, at the TreeAge Software web site:

http://www.treeage.com/treeagepro/manuals.htm

Web site resources

The support section of our web site includes a variety of information for users of the software:

http://www.treeage.com/support

http://www.treeage.com/resources

Example models

Both the trial and standard installations of TreeAge Pro includes numerous example decision trees and influence diagrams, in a sub-directory "TreeAge\TreeAge Pro\Tutorial Examples\".

Look for the "Tutorial Examples" shortcut in the START menu > All Programs > TreeAge Pro program shortcuts group.

E-mail and telephone support

If you are a registered user of TreeAge Pro with a current maintenance agreement, you are eligible for free e-mail and telephone support; limited technical support is available if you are evaluating the trial software. We would like to make your use or evaluation of the software as productive and pleasant as possible. If you need help, either:

- send an e-mail to support@treeage.com, or
- call us at 1-413-458-0104 (1-888-TREEAGE in US/Canada)

When requesting support, please specify your serial number, your problem or question, and which release of the TreeAge Pro 2007 software you are using (see About TreeAge Pro, under the Help Menu).

File compatibility

TreeAge Pro 2007 can open trees, influence diagrams, graphs, and other files created in previous versions of TreeAge Pro, as well as in DATA Pro or DATA. However, unless your TreeAge Pro 2007 software includes the Healthcare module, access to Markov and cost-effectiveness models created in either DATA or TreeAge Pro will be limited to viewing the file.

Trees, influence diagrams, and graphs created in TreeAge Pro 2007 are directly backward-compatible with TreeAge Pro versions. They are not directly backward-compatible, however, with DATA or DATA Pro. An Export command under the File menu can be used to export trees and influence diagrams in a format compatible with DATA Pro, DATA 4.0, or DATA 3.5. Certain features not present in the earlier software will be lost in the conversion and, in any event, care should be taken following conversion to confirm the accuracy of calculations.

Tables created by TreeAge Pro (*.tbl files) are backward compatible with DATA 4.0 and DATA Pro, but not with DATA 3.5. See Chapter 17 for more details.

> ***Note:***
> If you are familiar with our DATA or TreeAge Pro software, please refer to Appendix A in this manual for a description of new features and other changes.

Organization of this manual

This manual is for use with TreeAge Pro 2007. It includes basic tutorials for building and analyzing decision trees and influence diagrams in TreeAge Pro. It also provides a complete reference to all intermediate and advanced features found in the software, including those found in the Healthcare module and the Excel module.

Here is an overview of the user's manual contents:

Part I (Chapters 1–4) — For those unfamiliar with decision analysis, Chapter 1 provides basic background using a simple investment problem. Chapter 2 shows how to use TreeAge Pro to represent the investment model as a decision tree and then to perform expected value calculations. Chapter 3 shows how to build the same model as an influence diagram. Chapter 4 covers printing and exporting pictures of TreeAge Pro documents.

Part II (Chapters 5–11) — These chapters document many commonly-useful features when working with trees, including: building trees efficiently; creating and customizing graph windows; creating multi-attribute models; copying and pasting tree structure; and customizing the appearance of the tree.

Part III (Chapters 12–18) — This crucial section of the manual covers parameters, formulas, and various forms of sensitivity analysis. Defining uncertain parameters using variables is described in detail, as are tornado diagrams, multi-way sensitivity analysis, Monte Carlo simulation, distributions, and probabilistic sensitivity analysis.

Part IV (Chapters 19–24) — The chapters in this section cover a variety of intermediate and advanced software features. These include such topics as storing and "batching" complex analyses, creating complex cost formulas, linking to spreadsheets, using utility functions, and performing Bayes' revision.

Part V (Chapters 25–26) — The two chapters in Part V detail TreeAge Pro's powerful and flexible influence diagram interface. These chapters cover everything from the basics of influence diagrams to advanced topics such as asymmetry, Bayes' revision with multiple tests, expected value of perfect information, and conversion to decision trees.

EXCEL / COM MODULE

Part VI (Chapters 27–30) — These chapters provide information on: automating the creation of Excel charts and reports from analyses; integrating complex spreadsheet calculations and TreeAge Pro models; editing tables and variables using Excel; using the TreeAgeProLib and COM automation; and quickly importing existing spreadsheet models into an influence diagram.

HEALTHCARE MODULE

Part VII (Chapters 31–33) —The chapters in Part VII provide information on modifying existing decision trees for cost-effectiveness analysis. Baseline cost-effectiveness analysis, as well as sensitivity analysis and Monte Carlo simulation, are covered in detail.

Part VIII (Chapters 34–37) — The chapters in Part VIII provide information on creating and analyzing Markov models. Advanced Markov features and Monte Carlo microsimulation are covered in detail.

APPENDICES and INDEX

Following Part VIII are several appendices including a list of feature changes, and technical information on menus, preferences, functions, and other application details.

A comprehensive index for the manual follows the appendices.

Getting Started with TreeAge Pro 2007

The chapters in Part I provide background on decision analysis, decision trees, and influence diagrams. They include step-by-step tutorials designed to help new users learn how to use TreeAge Pro 2007 to build, analyze, and present decision trees and influence diagrams.

In this part:

- Chapter 1, "Decision Analysis Primer"
- Chapter 2, "A Decision Tree Tutorial"
- Chapter 3, "An Influence Diagram Tutorial"
- Chapter 4, "Printing and Exporting Pictures of Your Model"

Chapter 1

Decision Analysis Primer

This chapter introduces the core concepts and methods of decision analysis as implemented in TreeAge Pro 2007.

In this chapter:

- What is decision analysis? 8
- A simple problem: How should I invest $1,000? 10
- Decision trees 10
- Influence diagrams 13
- Further reading 14

What is decision analysis?

Decision analysis — as it is taught and practiced today in diverse fields such as oil and gas exploration, business, healthcare, law, and engineering — is a systematic approach to *decision making under uncertainty*. The process is designed to help decision makers think clearly about the many elements of complex decisions, such as:

- the range of possible consequences of actions (or *inaction*)
- preferences among different sets of consequences
- the impact of complex, unpredictable systems and processes (e.g., markets, geological structure, health)
- the actions of others (e.g., consumers, competitors, regulators, patients)

The concepts and methods of decision analysis are uniquely suited to incorporating into the decision making process both what is known about a problem, and also what is uncertain. TreeAge Pro is used in two of the key steps in decision analysis, modeling and analysis.

Modeling

Using decision analysis, a complex problem can be disaggregated into smaller problems and elements, which can be more readily comprehended. These components are then employed in building a model of the problem's essential elements.

First, a set of alternatives is compiled. Then, events and other factors that may affect the outcome of an alternative are identified. A factor whose impact on the final outcome is not known at the time of the decision is referred to as an uncertainty, and can be represented as either a structural element of the model, or a parameterized distribution used in a formula in the model. Based on the decision maker's objectives, one or more attributes are selected to quantify preferences for the range of possible final outcomes (and, ultimately, to rank alternatives). For example, a monetary scale would be used to measure different project cost scenarios and then to rank alternatives.

A particular strength of decision analysis, compared to basic spreadsheet analysis or statistical modeling, is the intuitive, visual form of the model. The model may be either a decision tree or an influence diagram, which are different means of visually representing the same problem.

A decision tree is a branching structure in which various node symbols are used to represent different kinds of events, including decisions and uncertainties, and a node's branches represent the outcomes or alternatives associated with that event. Every series of actions and outcomes is clearly represented with a distinct path.

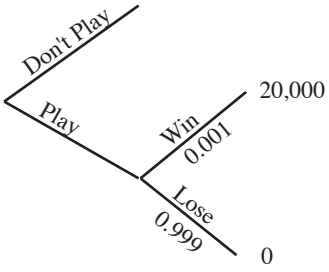
Alternatively, the same problem often may be represented more compactly by an influence diagram. In an influence diagram, each factor that directly or indirectly affects the final outcome is represented with a single node. Arcs between nodes show that one factor either influences another or precedes it in time.

Expected values and optimal decisions

While a model's explicit identification of the sequence and linkage of events is of great value in decision making, decision analysis is designed to do much more. Using basic concepts from probability theory and statistics, the decision maker can calculate a mean, or *expected*, value for each course of action. By calculating the value of each possible chain of events, and weighting uncertain results by the probability of each outcome, the decision maker can identify the sequence of decisions that will maximize value, minimize costs, or balance multiple attributes.

These calculations are commonly illustrated with a decision about playing a simple lottery, which can be thought of as a random variable having a simple probability distribution of outcomes:

> A $20,000 lottery offers a 1-in-1000 chance of winning. The expected value of playing this lottery — ignoring the ticket price — equals *0.001*(20,000)+0.999*(0)*, or $20. Based on expected value, a reasonable decision would be to purchase a ticket if it costs $20 or less.



Typically, decision analysis problems involve multiple uncertainties, with the outcome of the first lottery being a second lottery, and so on. The analysis of compound lotteries works backwards, calculating an expected value for the final lottery and using this result in the calculation of the prior lottery. When all lotteries have been resolved in this way, decisions are evaluated by optimization — picking the alternative with the best expected value.

Expected utilities and certainty equivalents

A decision maker's attitude towards *risk* can be incorporated into model calculations. Using the techniques outlined in utility theory, a choice between different lotteries can be made based on their expected utilities or certainty equivalents, which in turn depend on the decision maker's attitude towards risk, quantified using a utility function.

Sensitivity analysis and Monte Carlo simulation

As illustrated with the lottery example, one way in which decision analysis deals with uncertainty is to reflect it explicitly in the model's structure. Events which have a significant impact on outcomes, and which are not under the decision maker's control, can be described using chance nodes and incorporated into the model calculation. A problem may involve numerous uncertainties; not all of them can or should be represented in the structure of the model. To deal with this, deterministic sensitivity analysis and Monte Carlo simulation are used to examine the potential impact of parameter assumptions and other uncertainties.

Deterministic sensitivity analysis can take a variety of forms, including 1-, 2-, and 3-way sensitivity analysis and tornado diagrams; it can be used to identify critical uncertainties by examining the extent to which a model's calculations and recommendations are affected as a consequence of changing selected assumptions. Monte Carlo simulation, also referred to as probabilistic sensitivity analysis, can incorporate *all* parameter uncertainties. Probabilistic sensitivity analysis provides similar insights to deterministic sensitivity analysis, and can also quantify the level of confidence that can be placed in the model's results.

Sensitivity analysis, Monte Carlo simulation, and other analytical tools can also be used to improve decision making by determining the potential value of obtaining various kinds of information (*perfect*, *imperfect*, or *sampling information*) that might help resolve critical uncertainties.

A simple problem: How should I invest $1,000?

Now consider a simple example. You have $1,000 to invest, and two potential investments: a volatile equity investment, and a risk-free certificate of deposit (CD). You will reconsider your investment decision at the end of one year, but not earlier. You have no fees or taxes to consider, just an identical $1,000 investment in either case.

The CD pays simple interest at a rate of 5% annually — your return would be $50.

In researching the equity investment, you have come up with a simple probability distribution describing its year-end performance: 1) a 30% probability that its market value will have gone up by $500; 2) a 40% probability of a modest $100 increase in value; and 3) a 30% probability of a substantial drop in value, -$600. Your investment objective is to maximize growth, and you are sufficiently wealthy that the possible loss of $600 does not pose a material threat.

> ***Note:***
> Assigning 30%/40%/30% probabilities to the outcomes of the risky investment in the example follows a standard method for representing a probability distribution of outcomes based on expert opinion. This particular type of discrete distribution is referred to as a Swanson's mean, or 10/50/90, distribution. The three outcomes represent the 10th, 50th, and 90th percentile values elicited from the expert. A similar approach uses 25%/50%/25% probabilities for the three outcomes.

For a primer on applying the basic techniques of decision analysis to this problem, complete the rest of this chapter. If you are already familiar with the fundamentals of decision analysis, you may prefer to go directly to Chapter 2 or 3 to learn how to build this model in TreeAge Pro 2007 as either a decision tree or an influence diagram.

Decision trees

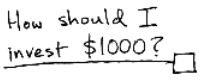
In building a decision tree, there are some basic guidelines to be considered:

- ***In the tree, events are ordered from left to right.*** The tree often follows a time ordering of events, as outcomes become known to the decision maker. Time ordering is only critical, however, when a decision is made prior to knowing the outcome of a different event or when the probabilities of one event are conditioned on another.
- ***Different kinds of events are distinguished using shapes called "nodes."*** A decision node (square) indicates a choice facing the decision maker. A chance node (circle) represents an event which has multiple possible outcomes and is not under the decision maker's control. A terminal node (triangle) denotes the endpoint of a scenario.
- ***Branches "sprouting" from a decision node represent the set of actions being considered.*** Decision alternatives are not required to be mutually exclusive (for example, one set of options could be "install smoke detectors," "install fire extinguishers," and "install smoke detectors and fire extinguishers").
- ***Branches from a chance node represent the set of possible outcomes of the event.*** The branches must be mutually exclusive and exhaustive — in other words, defined such that all possibilities are covered and none overlap. Their probabilities must sum to 1.0 (100%).
- ***Terminal nodes are assigned a value, referred to generically as a payoff***. All right-most nodes (those without branches) must be terminal nodes and have a payoff representing the net value — e.g., profit, cost, or utility — of that particular scenario's series of actions and events.

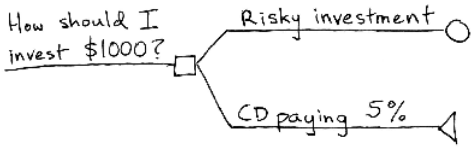
Using these guidelines, let's design (on paper) a decision tree that represents the investment problem posed above.

Tree structure

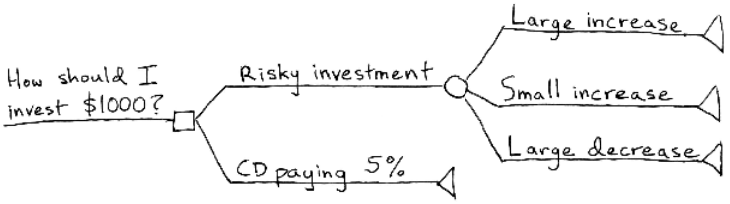
The first event, the decision between the available investment options, is represented by a square, decision node. This is the *root node* of the tree. It is labeled using a branch line to its left.



The two branches "sprouted" from the right side of the decision node represent the alternatives under consideration: (1) *Risky investment* and (2) *CD paying 5%*. Since there is no risk associated with the CD, this action really represents a final outcome, and a terminal node is drawn at the end of its branch. For the risky equity investment, however, we decided that there were three possible outcomes, so *Risky investment* is drawn as a chance node.



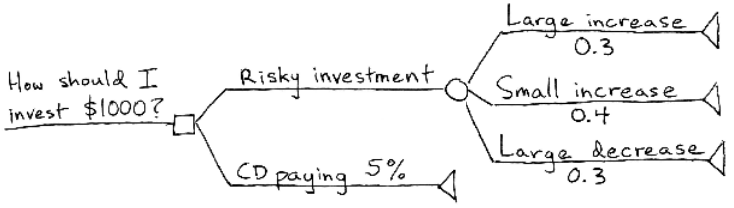
The three outcomes of the risky investment, *Large increase*, *Small increase*, and *Large decrease,* are drawn as branches emanating from the chance node. They are final outcomes, and are represented using terminal nodes.



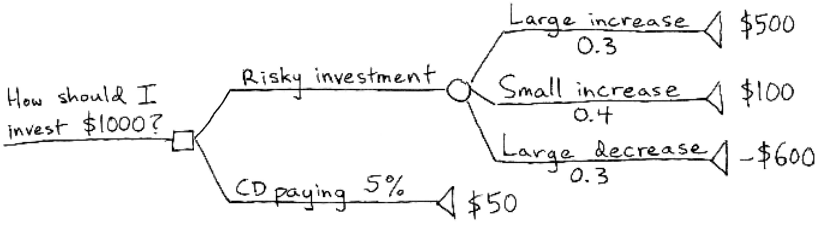
Probabilities and payoffs

Now the structure of the tree is complete. All that remains is to place the probabilities and payoff values in the tree.

Probabilities must be assigned to the branches emanating from a chance node, with the branch probabilities summing to 1.0. The probability for a particular outcome is indicated below the appropriate branch line.



Payoffs are assigned at every terminal node, and appear to the right of the node.



Calculating the tree

The investment decision tree is now complete, and can be evaluated. Decision tree calculations work backward, from right to left. Thus, calculating a tree is often referred to as "folding back" or "rolling back" the tree. The value of each node is determined as follows:

- The value of a terminal node is its assigned payoff value.
- The value of a chance node is its expected value. As described earlier, this is the mean value of the probability distribution specified by the chance node's branches.
- The value of a decision node is equal to its best option.

Now let's apply these rules to the investment tree.

The payoffs of the three terminal nodes are used. Working backward, starting from the topmost terminal nodes, you can find the expected value (EV) of the *Risky investment* chance node as follows:

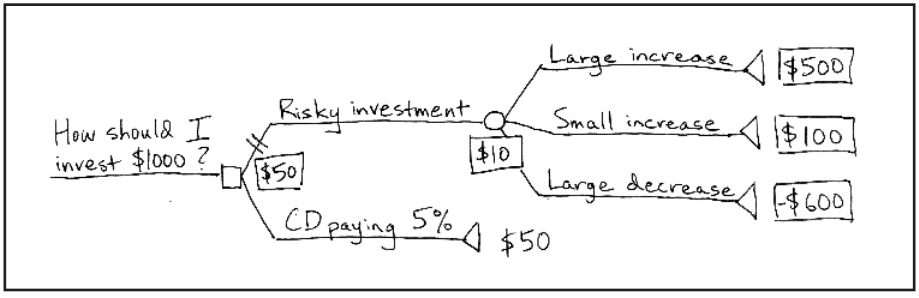
= (500 * 0.3) + (100 * 0.4) + (-600 * 0.3)

= 150 + 40 - 180 = 10.

All that remains is to calculate the value of the decision node — by comparing the expected values of the alternatives, and deciding which one is better. The expected value of *CD paying 5%* is $50, while the expected value of *Risky Investment* is $10. The CD offers the higher expected value and so it is optimal.

When a tree is rolled (or folded) back, the expected value of each node is drawn in a box to the right of the node. At decision nodes, the branches of *non*-optimal alternatives are identified by two slash marks.

Chapter 2 provides detailed instructions on how to use TreeAge Pro to build and calculate the simple investment decision tree.



Influence diagrams

Influence diagrams tend to be simpler on their face than decision trees. While they do not display the level of detail found in a tree (i.e., scenarios, probabilities, and payoffs), influence diagrams portray more clearly the factors to consider in decision making, and how those factors are related. Even in complex problems, where the decision tree is far too large to fit on a single printed page, the corresponding influence diagram is almost certain to be small enough for simple reproduction and efficient communication.

The design of an influence diagram is subject to a number of guidelines. Here are the basic ones:

- ***"Nodes" of different shapes represent the factors relevant to the problem.*** Each element of the problem — the final objective (e.g., maximizing profit), along with each decision, variable, and random event that can affect the objective — is represented by a single node. A *value* node (diamond) denotes a measure of the final objective. A *decision* node (square) is used to indicate a decision. A *chance* node (circle) is used to represent a variable (or event) whose value (or outcome) is unknown currently.
- ***Related nodes are connected by arcs.*** An arc ending in an arrow is drawn between two nodes to indicate that: (a) the first event precedes the second, and/or (b) the first event or action influences, or *conditions*, the second. An *influence* arc might indicate that the probabilities for one event depend on the outcome of a prior event or action. An influence arc might also indicate that an action or event makes some contribution to, or deduction from, the final objective (e.g., project cost, or profit).

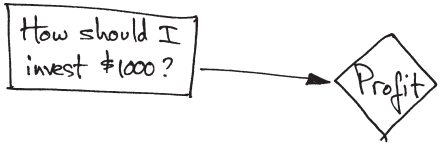
In accordance with these basic guidelines, we will design a simple, illustrative influence diagram that represents the same investment problem modeled using a decision tree in the previous section.

Influence diagram structure

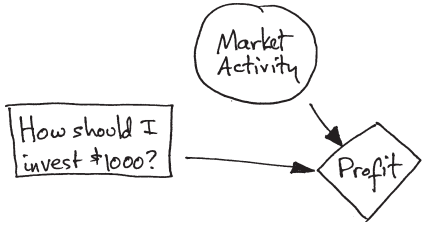
There are no strict rules for the order or manner in which you place the required nodes in the influence diagram. However, it may be helpful to start with the value node, which represents the overall objective — in this case, maximizing profit.



Next, you must consider what events and decisions may affect your return on investment. In this case, the first event to occur is the investment of the $1,000 investment. Your choice of investment will determine, at least in part, what the profit on the investment will be. So, add a square (or rectangular) node for the investment decision, and then draw an arc from the decision node to the value node ("Profit") to indicate the direction of influence.



In our simple model, the only unknown variable to consider is the fluctuation of the market. Thus, a circular (or oval) chance node is added to the influence diagram to represent the market risk, and an arc drawn from the chance node to the value node.



If the outcome of a chance event is known before a decision, then an arc must be drawn to indicate this. However, no arc is required when the opposite is true — as is the case in this model, a decision is assumed to precede all chance events, unless an arc indicates otherwise.

Analyzing influence diagrams

Based on the guidelines covered so far, the investment influence diagram is visually complete, and reflects the contours of the decision problem. However, much of the information visible on the face of a completed tree cannot be shown in the influence diagram — the investment outcomes, including their probabilities and values, and the investment options are not pictured.

In the influence diagram tutorial in Chapter 3, you will learn both how to build the visible influence diagram in TreeAge Pro, and then how to include the hidden information within nodes and arcs.

> ***Tip:***
> Influence diagrams can be very helpful in understanding and presenting complex decisions. Quantitative analysis in TreeAge Pro, however, requires an influence diagram to be converted into a decision tree. If you plan to analyze influence diagrams, read the sections on decision trees in this chapter, and both Chapters 2 and 3.

Further reading

For additional background on decision analysis in general, here are some suggested references:

- *Making Hard Decisions,* Clemen (1996), Wadsworth.
- *Decision Making and Forecasting,* Marshall and Oliver (1995), McGraw-Hill, Inc.
- *Decision Analysis: Introductory Lectures on Choices under Uncertainty*, Raiffa (1968), Random House.

There are also numerous books and journals dealing with the application of decision analysis in specific fields and industries. Here are a just a few selected references:

- *Decision Making in Health and Medicine,* Hunink, and Glasziou (2001), Cambridge University.
- *Decision Analysis for Petroleum Exploration, 2nd Ed.,* Newendorp and Schuyler (2000), Planning Press.
- *Coping with Risk in Agriculture*, Hardaker, Huirne, and Anderson (1998), CAB International.
- *Introduction to Decision Analysis: A Practitioner's Guide to Improving Decision Quality*, Skinner (1996), Probabilistic Publishing.

The TreeAge Pro installation includes numerous example decision trees and influence diagrams, including some examples from the above texts. The models are saved in a sub-directory "TreeAge\TreeAge Pro\Tutorial Examples\". Look for the "Tutorial Examples" shortcut under the START menu > All Programs > TreeAge Pro.

Visit the TreeAge Software web site for additional resources and useful links:

http://www.treeage.com/resources.htm

Chapter 2

A Decision Tree Tutorial

This chapter is designed to help new users of TreeAge Pro 2007 become familiar with the basic steps of using the software to build decision trees and perform simple calculations.

Healthcare module users: Refer to Chapter 31 for instructions on setting up cost-effectiveness decision trees, and Chapter 34 for instructions on working with Markov nodes and models.

In this chapter:

- The tree window 16
- Constructing a tree 17
- Entering payoff values 20
- Entering probabilities 22
- Setting calculation preferences 23
- Calculating the tree 24
- What's next? 25

The tree window

When you open TreeAge Pro, a new tree document is displayed below the menu and tool bars, as shown at the bottom of the page.

Two views of the tree are provided. The main tree view, with a white background, appears on the right and a compact, *Explorer view*, with a grey background, appears in the upper left. Below the explorer view is a *Properties grid*, displaying the label and other editable properties of the currently selected node.

As you will learn later in this chapter, each tree has its own set of display preferences. In addition, each tree has its own settings for the extra panes: the explorer view, properties grid, and debug pane. The size and position of the extra panes are saved with the tree.

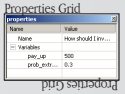
Each pane can be separately hidden, resized, or "docked." To hide or re-display a pane, choose the appropriate Hide… or Show… command from the bottom of the **Display** menu.

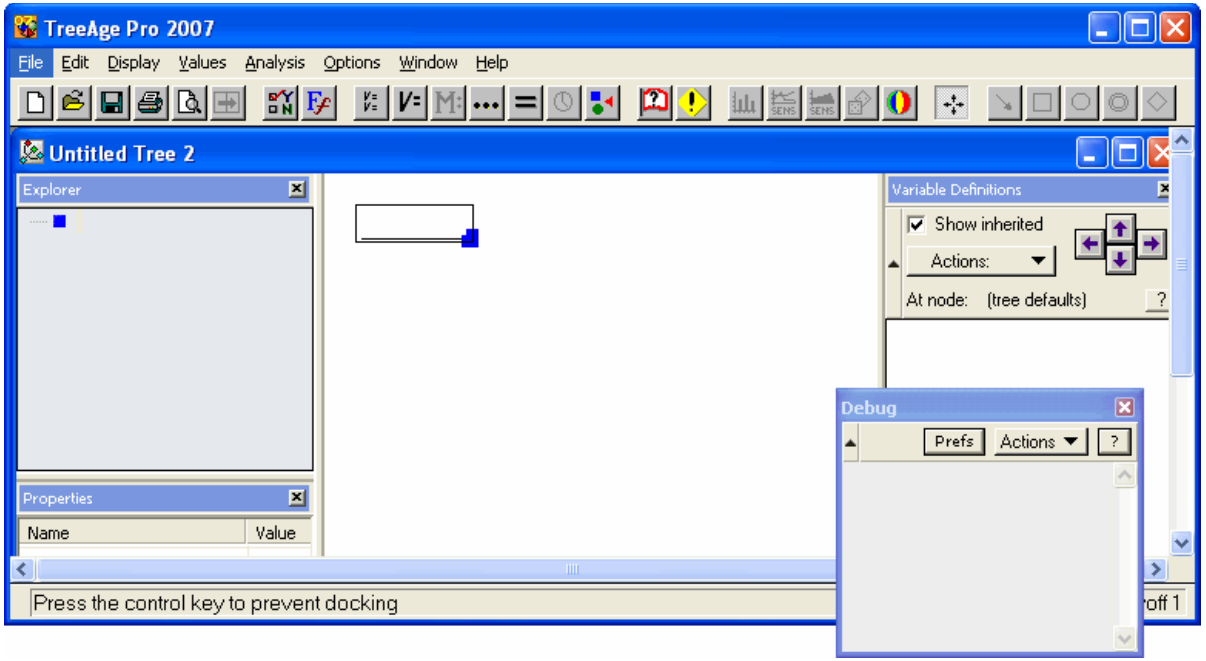
To resize a pane, use your mouse to click-and-drag the horizontal bar separating the two views, or the vertical bar separating them from the main tree view. The views can also be moved from their initial positions (docked on the left edge). Simply click-and-drag the title bar of one pane to a new position: floating inside or outside the TreeAge Pro window; or docked on the top, right, or bottom edge of the tree.

For details on using the extra panes to work more efficiently in very large trees, see Chapter 21.

The Properties grid:

In the sections of the manual on working with trees, most instructions ask you to click with the mouse in the main tree view. However, most navigation can also be accomplished in the Explorer view, and many text and value editing tasks illustrated in the tutorial using menus and dialogs can also be done in the Properties grid. Throughout the manual, a small **Properties grid icon** like the one shown here is used to highlight node editing tasks that can be done in the properties grid.





Tutorial Notes:

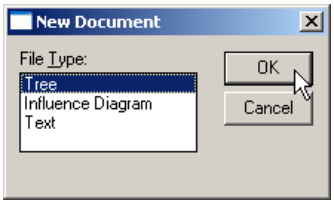
- A word in **boldface** followed by the > symbol indicates a menu selection. For example, **File** > New… refers to the File menu's New… command. Other words in boldface are labels of buttons.
- Words to be typed by you as part of the tutorial are shown in Courier typeface:
  `"How should I invest $1000?"`
- Some procedures explained in this chapter are not required steps for building the investment decision tree. Lines starting with the ⇨ character are required tutorial steps; other instructions start with the ▷ character.

Constructing a tree

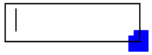
The tutorial in this chapter is based on the investment decision problem described in Chapter 1. If necessary, create a new, empty tree now.

► **To create a new tree:**

▷ Choose **File >** New…, select Tree from the list of document types, and click OK (or press the ENTER key).



Each new decision tree document starts with a single decision node — a blue square, with a line and text box to its left. This is the *root node* of the tree. Although the root node cannot be deleted, you can change it to a different type of node, when necessary. A litigation model, for example, might have a chance node as the root node.



Selecting and deselecting a node

In a new tree, the root node is initially selected. TreeAge Pro shows that a node is selected by filling in the node symbol.

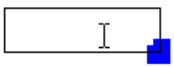
► **To deselect a node using the mouse:**



▷ In the tree window, click in the empty white space anywhere away from the node and its branch line.

When a node is not selected, its symbol is hollow.

► **To select a single node using the mouse:**



▷ Click directly above the line to the left of the node. (When positioned correctly, the pointer changes to a text caret.)

If you *double-click* on the node symbol accidentally, it will sprout two branches.

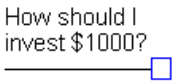
Entering node text

When a single node is selected, a box and a blinking text insertion caret appear above the line to the left of the node. You can enter a brief phrase in the box to describe the event — in this case, the investment decision.

► **To enter a name or short description for a node:**

⇨ Select the root node.

⇨ Type `How should I invest $1000?` in the text area. Press ENTER after the word "I" to insert a carriage return.



⇨ When you are done, click outside the node to deselect it.

⇨ If you click on the node again, and the properties grid is visible, it can also be used to make changes to the node label.

If the text appears too small on the screen, one way to fix the problem is by zooming in.

► **To zoom in (or zoom out):**

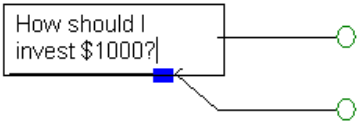
▷ Choose **Display** > Zoom In (or Zoom Out).

Adding branches (and nodes)

Branches must be added to the decision node to represent the available investment choices.

► **To add branches:**

⇨ Select the root node, and choose **Options** > Add Branches.



Two branches are added to the node, each ending in a chance node. Once you have two branches, repeating the Add Branch(es) command will add additional branches, one at a time.

Another way to add branches to a node is to double-click on the node symbol. When the mouse pointer is positioned over the node's symbol, the cursor will change to a branch cursor (shown at left) to indicate that double-clicking will add branches.



Deleting branches

If you add an extra branch by mistake (for example, if you double-click on a node when trying to select it, instead of just clicking), there are a couple of ways to fix the tree.

► **To undo the previous action or actions:**

▷ Choose **Edit** > Undo [*last action*].

Or, if you do not want to undo other actions performed *after* mistakenly adding the extra branch, you can select a branch and delete it.

► **To delete a branch:**

▷ Select the branch to remove and choose **Options** > Delete Branch.

Navigating the tree using the keyboard

In addition to using the mouse to select a particular node, it is also possible to use the keyboard to change the selected node.

Even if no node is currently selected, there is a keyboard shortcut that will select the root node.

► **To select the root node:**

▷ Hold down the CONTROL key and press the HOME key.

To move the selection from the current node to an adjacent node, use the arrow keys. Try using the arrow keys to select the top branch.

► **To move the selection one node to the right:**

▷ Press the right arrow key.

Now, use the actions you have learned to complete the tree structure.

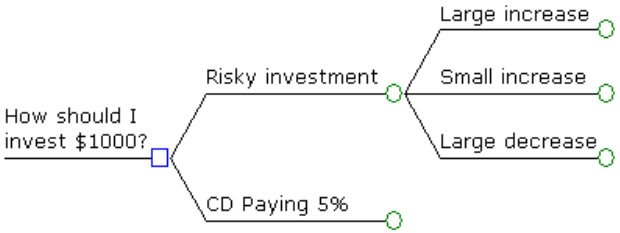
⇨ Select the top branch of the decision node, and type in its name, `Risky investment`.



⇨ Add three branches to Risky investment, and name them `Large increase`, `Small increase`, and `Large decrease`.

⇨ Select the bottom branch of the decision node, and name it `CD paying 5%`.

The tree should now look like this:



Tree-building tips:

- Rather than entering extensive comments in the node itself, it is often better to use note boxes or node comments (see Chapter 10).
- Most tree-building commands are found under the **Options menu**; see Chapter 9 for an overview of these commands. (Some commonly-used commands can also be accessed by right-clicking on a node.)
- If you delete a branch that is a parent (i.e., that has branches), its subtree (i.e., its children as well as their descendants) will not be deleted; instead the children and their descendants will move up one generation. See Chapter 9 for more information on inserting and deleting nodes.
- To move the text cursor left or right within node text, hold down the CTRL key while pressing the arrow keys. Toggling the **Navigation tool bar button**  (shown at right) will reverse this behavior (i.e., using the arrow keys alone will move within node text, while using the arrow keys with the CTRL key will move the node selection). See Appendix B for details.

Saving the tree

Now that all necessary branches have been added, it is a good time to save your work.

Models you create in TreeAge Pro are documents. You save, open, and close trees and other documents in TreeAge Pro the same way you do in other programs — using the File menu commands.

► **To save a TreeAge Pro document:**

⇨ Choose **File** > Save…, or click the  icon on the tool bar.

⇨ In the Save As dialog, select or create a directory, type `Stock Tree` for the file name, and press ENTER or click Save.

Tip:

By default, TreeAge Pro 2007 periodically auto-saves copies of files that you open. You can adjust the backup/autosave settings under Edit > Preferences…. In case of a problem, you can recover the latest auto-saved copy; see Appendix C for details. You should still save your work periodically, however. It is also recommended that you save your tree files in a location that is periodically backed up, such as a documents or projects directory on your computer or a network shared drive.

Entering payoff values

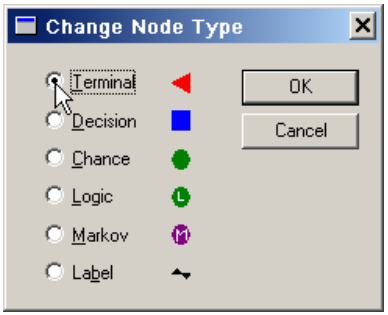
Note that each branch you create ends, by default, in a chance node. If a node represents a path endpoint, you must change it from a chance node to a terminal node in order to assign a payoff value.

► **To change an endpoint to a terminal node:**

⇨ Select a node with no branches — in this case, Large increase.

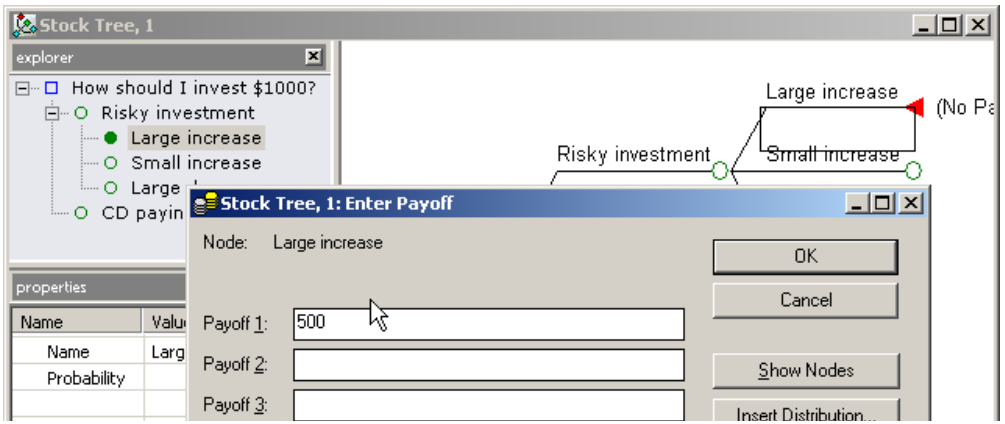
⇨ Choose **Options** > Change Node Type…, or click on the Change Node Type tool bar button.





⇨ In the Change Node Type dialog, click on Terminal, and press ENTER or click OK.

After changing *Large increase* to a terminal node, TreeAge Pro automatically opens an Enter Payoff window for the node. Use the Enter Payoff window to assign the numeric payoff for the *Large increase* terminal node.



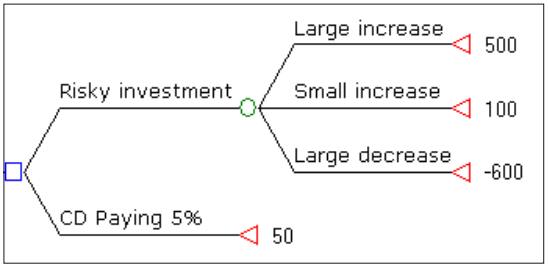
► **To assign a payoff:**

⇨ In the Enter Payoff window for Large increase, type `500` for Payoff 1, and press ENTER or click OK.

The payoff should be displayed in the tree window to the right of the new terminal node.

⇨ Follow the same steps for the nodes Small increase, Large decrease, and CD paying 5% — change the node type to terminal and then enter the payoff (100, -600, and 50, respectively).

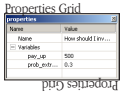
Your investment decision tree should now look like this:



If you find that you have made a mistake in entering payoffs, you can open the Enter Payoff window again.

► **To change the payoff at an existing terminal node:**

▷ To re-open the Enter Payoff window, select the node and choose **Values >** Change Payoff… (or double-click on the terminal node symbol). If the properties grid is visible, it can be used to edit the selected node's payoff, instead.



Notes on payoffs:

- The investment decision tree, and most other examples in this manual, require only Payoff 1. However, each tree can use up to nine different payoffs, or attributes; see Chapter 7 for details.
- See the Healthcare module documentation, starting with Chapter 31, for details on setting up cost-effectiveness calculation method trees.
- Chapter 11 and Appendix C describe how to customize the appearance of payoffs and other visual elements of trees in TreeAge Pro.
- Chapters 12 and 14 will show you how to use variables and formulas in payoffs.

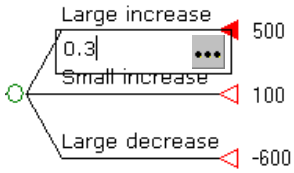
Entering probabilities

Now, the probabilities must be entered for the three possible outcomes of Risky investment. To enter a probability for a chance node's branch, click *below* the branch line. Or, if the branch is already selected, pressing TAB will switch the text cursor from above the branch line to below the line, and vice versa.

► **To enter probabilities:**

⇨ Select the Large increase terminal node, and press the **TAB** key to edit the probability. Type `0.3`, and then press TAB to save the probability. (Or, if the properties grid is visible, it can be used to edit the node's probability.)





⇨ Follow the same procedure to enter probabilities of 0.4 for Small increase and 0.3 for Large decrease.

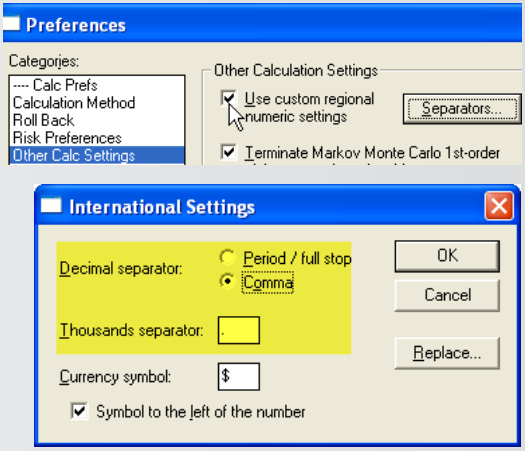
Notes:

- TreeAge Pro 2007 provides visual cues when the mouse pointer is over a node's probability box or branch label.
- The ellipsis button (…) will open an editor where complex expressions using variables and functions can be easily entered.
- The hashmark (#) can be used in place of a probability expression for one branch, in order to have TreeAge Po automatically calculate the complement during calculations.

IMPORTANT!

Regional (e.g., European) numeric settings:

- If your computer is set up to use commas (",") to represent decimals, rather than periods ("."), you should enter numbers in TreeAge Pro this way. Normally, you will enter numbers in TreeAge models just as you would in a spreadsheet or calculator.
- If your model will be shared by users with different regional numeric settings, use a special tree preference to instruct TreeAge Pro to override a computer's regional settings (i.e., reverse the usage of separators) for that particular tree. This setting is found in the Other Calculation Settings category. See Chapter 12 and Appendix C for details.



Setting calculation and numeric formatting preferences

The investment example's tree structure is complete and all values required for calculation have been entered. Before you evaluate the tree, however, a few basic preferences should be specified.

Most importantly, TreeAge Pro needs to know how to select an optimal path at decision nodes. For the investment tree, a strategy that maximizes your income is preferred. You can also specify the appropriate numeric formatting to use when displaying calculated values (number of decimal places, currency symbols, and abbreviations).

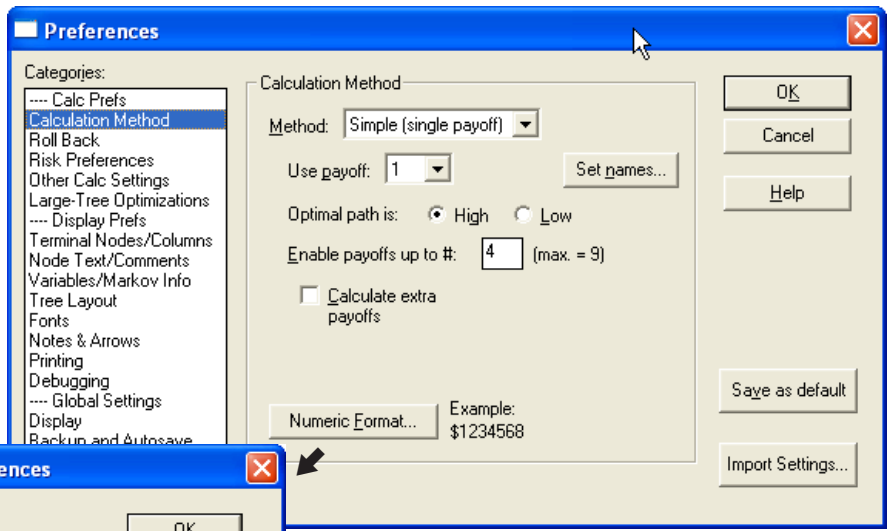
► **To set the calculation method and numeric formatting:**

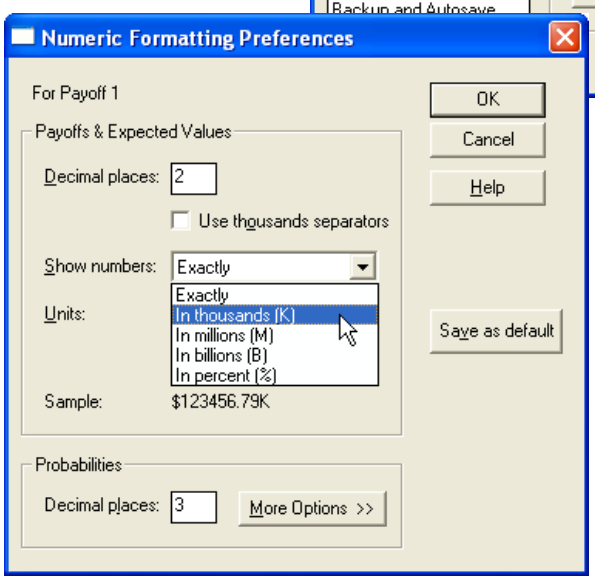
⇨ Choose **Edit** > Preferences…, or click the tool bar button.



⇨ The Calculation Method settings should be Simple, using payoff 1, with the optimal path set to High, for maximization.

⇨ Confirm that the example value shown next to the **Numeric Format…** button uses appropriate formatting. If changes are required, click the button to change these Numeric Formatting preferences.





Each tree has its own independent set of preferences. Changes to this tree's preferences will not affect other trees unless you check the "Save all settings as default" option before clicking OK, which will update the default settings used for new trees.

⇨ Press ENTER or click OK to apply changes, and save the tree again now, by choosing File > Save.

See Chapter 5 and Appendix C for more details on numeric formatting and other tree preferences, including information on importing preferences from other trees.

Calculating the tree

Now the tree should be ready for analysis. Start by rolling back the tree, which performs the expected value calculations described in Chapter 1.

► **To roll back the tree:**

⇨ Choose **Analysis >** Roll Back.

The rolled-back tree should look essentially like the picture below.

If TreeAge Pro generates an error message, read it to find out what needs to be fixed. Possible problems include endpoints that are not terminal nodes, and missing probability or payoff values.

If roll back works, but reports look different than those shown below, you may need to check your work, and fix any probability or payoff that was entered incorrectly.

Or, perhaps you have not specified appropriate numeric formatting, as described on the previous page.

Changes cannot be made to structure or values while Roll Back is on (look for a check mark next to the Analysis > Roll Back menu item).

► **To turn off roll back display:**

⇨ Choosing **Analysis >** Roll Back again will turn off roll back, and allow you to make changes to your tree.

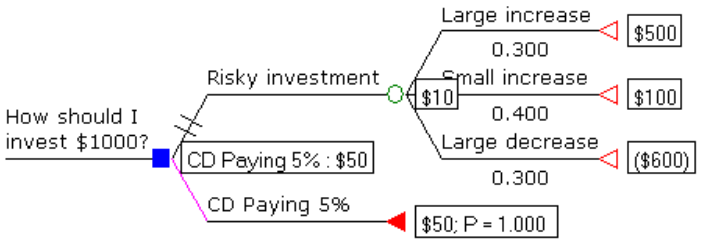
Roll back text report

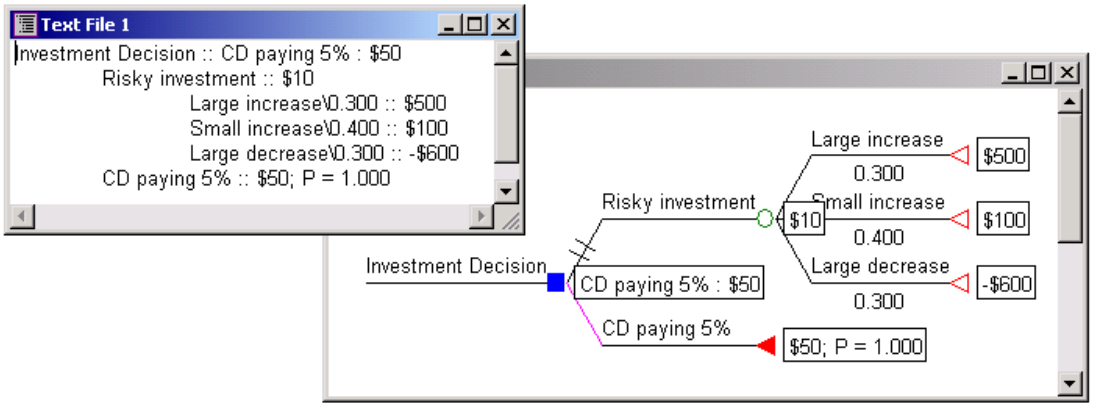
It is possible to export a text representation of a rolled-back tree. Similar in structure to an outline of a tree (see Appendix F), a roll back report also displays the calculated values for all nodes.

► To create a roll back report:

▷ Roll back your tree, and choose **File >** Convert to Rollback Report.

The roll back report, shown at the bottom of the page, can be printed, saved, or copied into a spreadsheet.





What's next?

This completes the basic decision tree tutorial. You are now ready to use TreeAge Pro to build your own decision trees.

The tutorial in the next chapter shows how to build the same investment model, but starting from an influence diagram.

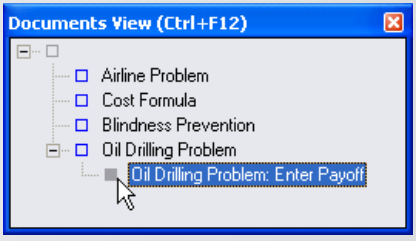
Some questions you may have at this point:

- **How can I build a tree to calculate cost-effectiveness?** The cost-effectiveness form of multi-attribute calculations is a feature available in the TreeAge Pro Healthcare module. For a tutorial on cost-effectiveness models, see Chapter 31.
- **How can I include my model in a presentation?** See Chapter 4, Printing and Presenting Trees and Influence Diagrams.
- **How can I do sensitivity analysis?** See Chapter 12 on variables and sensitivity analysis.
- **How can I generate a risk profile histogram?** See Chapter 5 (section on probability distributions).
- **What if I have very complex cost formulas?** See Chapter 14 on building formulas using variables, for example.
- **How can I assign a utility function?** See Chapter 24 on utility functions and risk preference.

You might take some time now to review the topics in these chapters, especially those in Part II, "Working with Decision Trees."

As you work with the software and have questions about functionality, bear in mind that the table of contents at the front of the manual and the index at the rear will simplify finding answers.

Documents View – The TreeAge Pro 2007 window adds a floating window showing a "tree" depicting all open documents, as well as other active windows open for editing variables, tables, and other formulas/expressions. Clicking on a "node" in the Document View will bring up the selected window. The Documents View window can be resized, and closed/reopened (from the Window menu).



Chapter 3

An Influence Diagram Tutorial

The tutorial in this chapter is designed to help new users of TreeAge Pro become familiar with the basic steps of using the software to build an influence diagram. Chapters 25 and 26 cover intermediate and advanced influence diagram topics.

Users of the Excel module should refer to Chapter 29 for special instructions on automatically converting a spreadsheet into a deterministic influence diagram.

If you plan to use TreeAge Pro to build only decision trees, you can proceed to Chapter 4.

In this chapter:

- Constructing an influence diagram 28
- Entering hidden information 32
- Calculating the model 38
- What's next 39

Constructing an influence diagram

The tutorial in this chapter explains in detail the software commands needed to build an influence diagram model of the investment decision described in Chapter 1. If you have questions while working through the tutorial, you may find it helpful to refer to the original explanation of the problem and the guidelines for building influence diagrams, in Chapter 1.

Tutorial Notes:

- A word in boldface followed by the > symbol indicates a menu selection. For example, **Diagram** > Change Node Type… refers to the Diagram menu's Change Node Type… command. Other words in boldface are labels of buttons.
- Words to be typed by you as part of the tutorial are shown in Courier typeface:
  `"How should I invest $1000?"`
- Some procedures explained in this chapter are not required steps for building the investment decision tree. Lines starting with the ⇨ character are required tutorial steps; other instructions start with the ▷ character.

The influence diagram window

If you completed the tutorial in Chapter 2 on building a decision tree, the Stock Tree window may still be open and active. The tree document does not need to be open while you work on the influence diagram tutorial.

► **To close a file:**

▷ Choose **File >** Close.

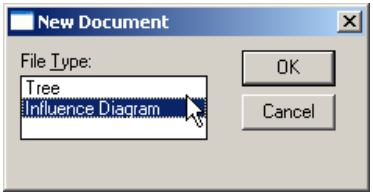
▷ If you made any changes to the tree which you have not saved, TreeAge Pro will prompt you to do so.

Now, to get started, you will need a new influence diagram document.

► **To create a new influence diagram:**



⇨ Choose **File >** New.



⇨ Click on Influence Diagram in the list of file types, and press ENTER or click OK.

Creating and selecting nodes

Unlike a new tree, which starts with a root node, a new influence diagram window is completely blank. The first step in building the diagram is to add the required nodes. As in the influence diagram primer in Chapter 1, let's start by showing the final objective — profit.

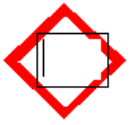
► **To add a node to an influence diagram:**

⇨ Click on the tool bar button that represents the kind of node you want — in this case, the red diamond, for a value node.



⇨ Click somewhere in the influence diagram to place the node.

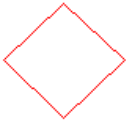
You will see a new, selected value node. TreeAge Pro indicates that the node is selected by showing it in thick outline. Now, see what the node looks like when not selected.



► **To deselect a node using the mouse:**

▷ Click in the influence diagram window anywhere away from the node.

You can tell the node is no longer selected because its outline is now thin.



Now, reselect the node so that you can enter a text label.

► **To select a node for text editing:**

▷ Click in the white space inside the node. (When positioned correctly, the mouse pointer should change to a text cursor.)

When a node is selected in this fashion, a blinking text cursor and text box will appear inside the node.

Entering node text

You can enter a word or brief phrase in the text box to describe this element of the problem — in this case, the investment objective.

► **To enter a name or brief description for a node:**

⇨ Click inside the value node to select it for text editing.

⇨ Type `Profit` in the text area.

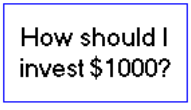
⇨ When you are done, click outside the node to deselect it.



Following the same steps as in Chapter 1, now add a node for the first event that influences return on investment — the decision.

⇨ Create a decision node (blue square). 

⇨ For its label, enter `How should I invest $1000?`, pressing ENTER after the word "I" to insert a carriage return.





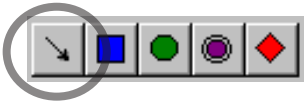
Adding arcs

In the influence diagram primer in Chapter 1, arcs were introduced as a means of displaying the relationships between actions, variables, events, and objectives. The direction of influence between the two nodes added so far is from the investment decision to Profit, so an arc should be created that points to the value node.

There are three ways to create an arc between two nodes using the mouse. One easy method uses the arc button on the tool bar.

► **To draw an arc:**

⇨ Click on the arc button on the tool bar. Now, when you move the mouse pointer over a node, TreeAge Pro will highlight that node.



⇨ Using the mouse, click and drag from the influencing node to the conditioned node — in this case, click on How should I invest… and drag to Profit.

⇨ Release the mouse button to create the arc.



You should now have an arc pointing from the decision node to the value node. The square "handle" on the arc shows that it is selected. If your have made any mistakes, skip to the section on editing nodes and arcs.



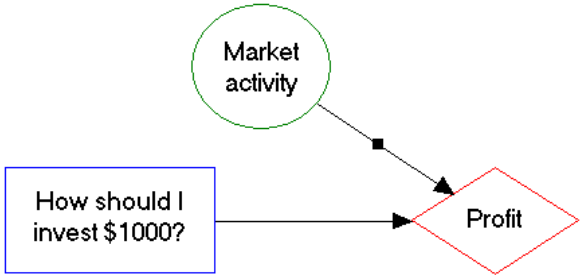
Now, add the other required node — a chance node representing the risk inherent in choosing the stock — and its influence arc.

⇨ Create a chance node (green circle), and for the node name enter `Market activity`.



⇨ Create an arc from Market Activity to Profit.

Here is essentially how your three-node influence diagram should look now:



Now that all nodes and arcs have been added, take a moment to save your document. You save, open, and close influence diagrams and other documents in TreeAge Pro the same way you do in other programs — using the File menu commands.

► **To save a TreeAge Pro document:**

⇨ Choose **File** > Save, or click the icon on the tool bar.



⇨ In the Save As dialog, select or create an appropriate directory, type `Investment Decision` for the file name, and press ENTER or click Save.

TreeAge Pro does perform auto-saves, based on your preferences (see Edit > Preferences..., Backup and Autosave). However, you should still save your work periodically, as you do when working in other document software.

Arc shortcuts

- Another way to create an arc is to **right-click** on the influencing node, choose Draw New Arc from the quick menu, and then click on the conditioned node to complete the arc.
- The quickest way to create an arc is to **control-click and drag** from the influencing node to the conditioned node.

Editing existing nodes

If you place the wrong type of node, instead of deleting the node and creating a new one, you can change its type (e.g., from decision to chance).

► **To change a node's type:**

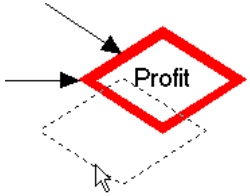
▷ Select the node by clicking anywhere on or in the node, and choose **Diagram >** Change Node Type….

▷ Click on the correct node type and press ENTER or click OK.

There are no strict rules about placement of influence diagram nodes. If nodes overlap, however, you can rearrange them to achieve better spacing.

► **To move a node:**

▷ Click on (or just inside) the node's outline, and drag it to a new location.



If you add an unnecessary node by accident, you can remove it.

► **To clear (delete) a node:**

▷ Select the node by clicking once on its outline, and choose **Edit >** Clear.

Selecting and editing existing arcs

To change or remove an arc, you must be able to select it.

► **To select an existing arc:**

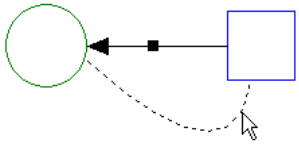
▷ Click on, or very close to, the arc — either the line or its arrow. A selected arc will display a square handle.



Instead of moving a node out of the way of an arc, it is possible to make the arc curve around the node.

► **To curve an arc:**

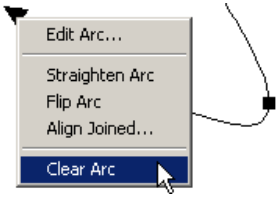
▷ Select the arc. Click and drag the arc's square handle to create the curve. Release the mouse button when you have the desired curve.



If you draw an arc between the wrong two nodes, you can remove it.

► **To clear (delete) an arc:**

▷ Select the arc and choose **Edit >** Clear Arc. Or, right-click on the arc and choose Clear Arc from the quick menu.



If you draw an arc in the wrong direction, you have the option of flipping the arc instead of deleting it.

► **To flip an arc:**

▷ Select the arc and choose **Diagram >** Flip Arc.

Entering hidden information

TreeAge Pro extends the capabilities of the standard influence diagram by adding information *inside* nodes and arcs — including lists of event outcomes, probabilities and payoff values. This hidden information is used to convert the influence diagram into a fully-configured tree, ready for analysis. If you do not plan to convert the influence diagram to a tree for analysis, then you may skip this section

Assigning alternatives and outcomes

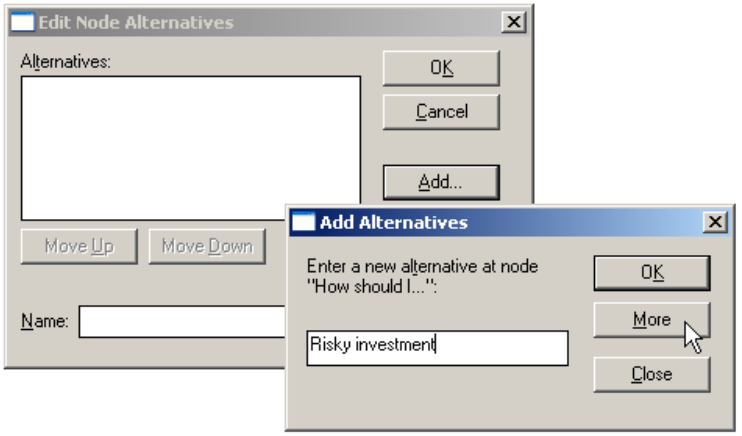
Before you convert an influence diagram into a tree, all decision nodes must have a list of alternatives, and all chance nodes must have a list of all possible outcomes. These lists correspond to the branches of the decision or chance node in the tree.

► **To enter a list of alternatives at a decision node:**

⇨ Double-click on the outline of the node (or select it and choose **Diagram >** Alternatives, or right-click on the node).

⇨ In the Edit Node Alternatives dialog, click the **Add…** button.

⇨ In the Add Alternatives dialog, enter the name of the first alternative, `Risky investment`, and click More.



⇨ Enter the name for the second and last alternative, `CD paying 5%`, and then press ENTER or click OK to close the Add dialog.

⇨ Review the list of alternatives in the Edit Node Alternatives dialog. If it looks correct, click OK to return to the influence diagram window.

Now enter the list of possible outcomes at the Market activity node.

► **To enter a list of outcomes at a chance node:**

⇨ Double-click on the outline of the node, or select it and choose **Diagram** > Outcomes.

⇨ In the Edit Node Outcomes dialog, click the **Add…** button.

⇨ In the Add Alternatives dialog, enter the name the first outcome, `Large increase`, and click More. Repeat for the second and third outcomes, `Small increase` and `Large decrease`. Press ENTER or click OK to return to the list of outcomes.

⇨ If the list looks correct, click OK.

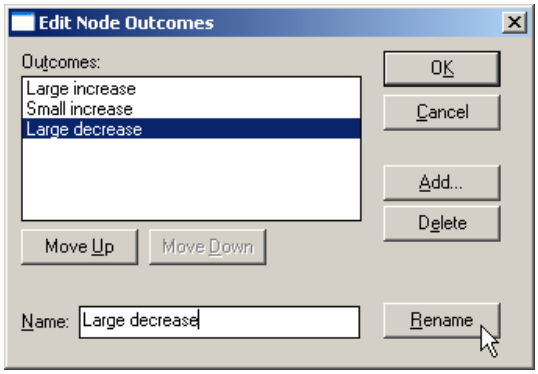
If there is a problem with an item in the list of alternatives or outcomes, use the buttons in the dialog to make changes.

► **To delete an alternative or outcome:**

▷ Select the item to remove from the list, and click the **Delete** button.

► **To rename an alternative or outcome:**

▷ Select the item to rename in the list, retype the name in the Name text edit box, and click **Rename**.



► **To reorder alternatives or outcomes:**

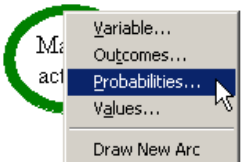
▷ Select the item to move up or down in the list order, and click the **Move Up** or **Move Down** button.

Entering probabilities

One you have assigned outcomes to a chance node, you can enter their probabilities.

► **To enter probabilities at a chance node:**

⇨ Select Market activity and choose **Diagram** > Probabilities. Or, right-click on the node and choose Probabilities from the quick menu.



Probabilities and other values associated with influence diagram nodes are entered using a "mini-tree" view.

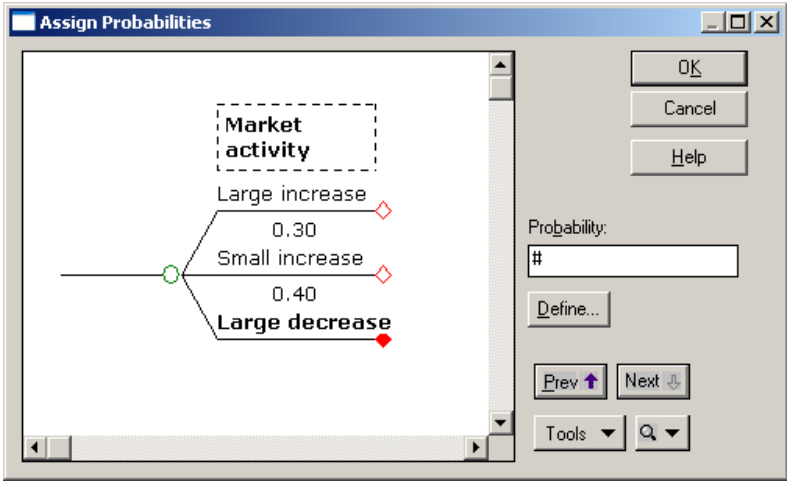
The mini-tree is created based on the lists of outcomes and alternatives found at the selected node and any nodes that *influence* it. The Market activity node has no conditioning events, so only its own outcomes are shown in the mini-tree.

Each node in the mini-tree that requires your attention will have a red diamond symbol. The active node will have its red diamond filled in and its name in boldface.

⇨ Enter the probabilities for Large increase and Small increase as `0.3` and `0.4`, respectively. Click in the mini-tree, or use the **Next** button or the down arrow key, to move to the next node.

⇨ TreeAge Pro automatically fills in the last branch with its remainder symbol, the hashmark ("#"). Either accept this, or type in a `0.3`, if you prefer to see a numeric value.

⇨ Press ENTER or click OK to return to the influence diagram.



Entering values

Since there are only a small number of possible scenarios in the investment example, the easiest approach for this model is to enter the numeric value (i.e., profit or loss) for each possible scenario, rather than creating a formula to calculate payoffs. Although the manual refers to these values generically as payoffs, they can be costs or any other attribute.

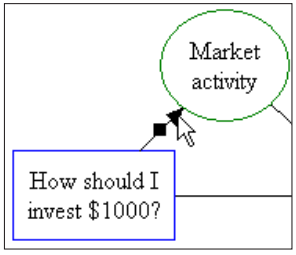
► **To enter/edit a list of payoffs at a value node:**

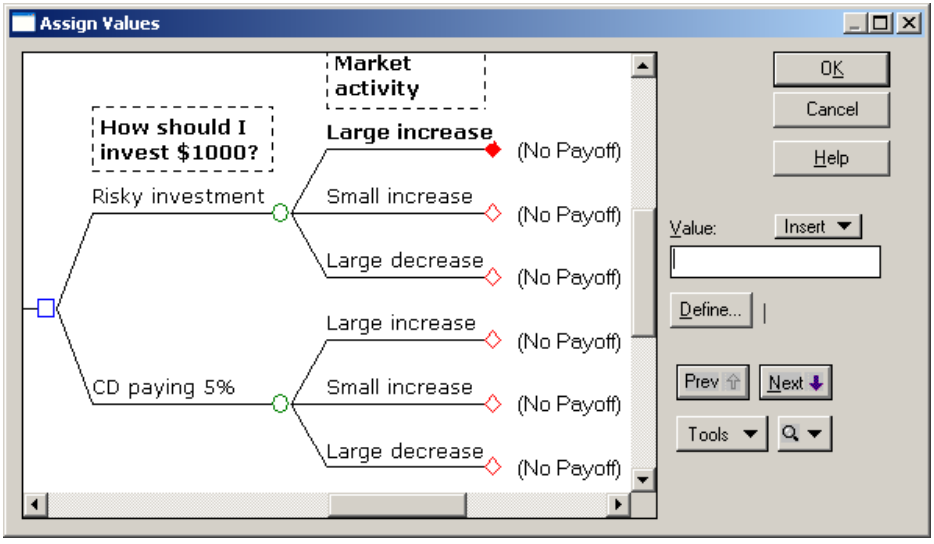
⇨ Double-click on the Profit node, and then click the **Values…** button. Or, select the Profit node and choose **Diagram >** Values.

As with assigning probabilities, a mini-tree is used for assigning payoffs. The mini-tree is created based on the lists of outcomes and alternatives found at the chance and decision nodes that influence the value node.

Note that the conditioning nodes/events How should I invest… and Market activity both appear in the mini-tree, based on the arcs you have drawn. The model does not yet reflect the fact that market fluctuations are irrelevant if you decide to invest in a CD.

Entering the same $50 payoff value at each CD paying 5% endpoint would allow us to work around the problem. A better solution, however, is to get rid of the unnecessary branches. This *asymmetry* can be created by adding a special arc between the decision node and the chance node.





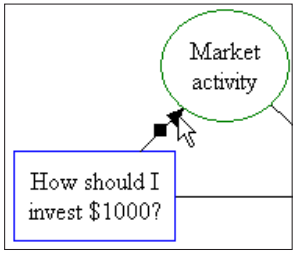
Creating asymmetry: arc information

Start by creating an arc from the decision node, which causes the asymmetry, to the chance node.

⇨ Close the mini-tree and Edit Node Variables dialogs.

⇨ Click on the arc button on the tool bar. Click and drag from the decision node to the Market activity node. Release the mouse button to create the arc.





An arc, like a node, can contain hidden information — in particular, an arc between any two chance and/or decision nodes.

► **To create asymmetry using an arc:**

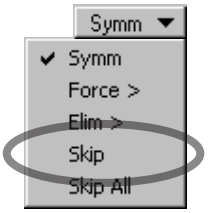
⇨ Double-click on the arc you just created, from the decision node to the chance node. Or, right-click on the arc and choose Arc Info….

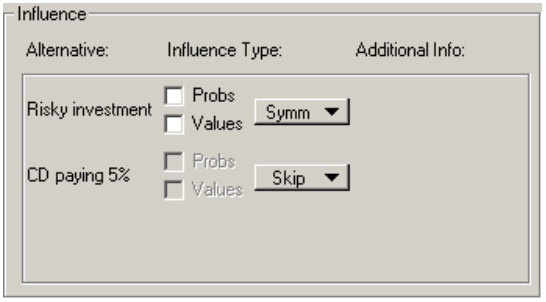
The Arc Data dialog shows numerous options. First, turn your attention to the lower section, labeled Influence.

Each alternative in the *conditioning* node has a separate group of influence settings. Since the *conditioned* node is a chance node, settings include probabilistic and/or value influence (the check boxes) and structural influence (the pop-up menu, currently reading Symm).

⇨ Uncheck all four boxes, to indicate that your decision has no value or probabilistic influence on market activity.

⇨ For CD paying 5%, change the symmetry menu to Skip.

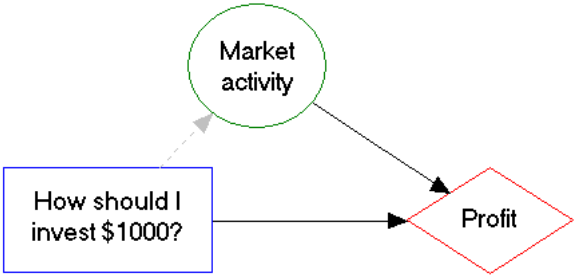




⇨ Click OK to close the Arc Info dialog.

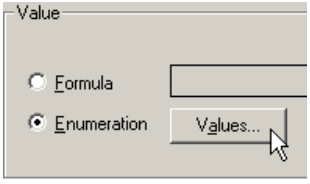
Entering values, revisited

This arc now indicates to TreeAge Pro that, in the CD paying 5% path, the Market Activity event should be skipped for structural and analytical purposes. Because there is no probabilistic or value influence through this arc, TreeAge Pro uses a dotted gray line for the arc.

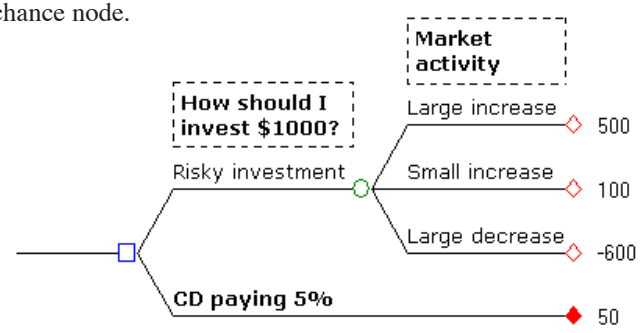


Verify that this change has had the desired effect on the enumeration of endpoints requiring payoffs at the Profit node.

- Double-click on the Profit node, and click **Values…**. Or, select Profit and choose **Diagram** > Values.



The mini-tree should now terminate the CD paying 5% path. If not, go back and check the arc information for the influence from the decision node to the chance node.



- Enter `500` for the payoff of Risky investment followed by Large increase, `100` for Risky investment and Small increase, `-600` for Risky investment and Large decrease, and `50` for CD paying 5%.
- Click OK twice to return to the influence diagram.

Before going on, take a moment to save the changes you have made to the influence diagram.



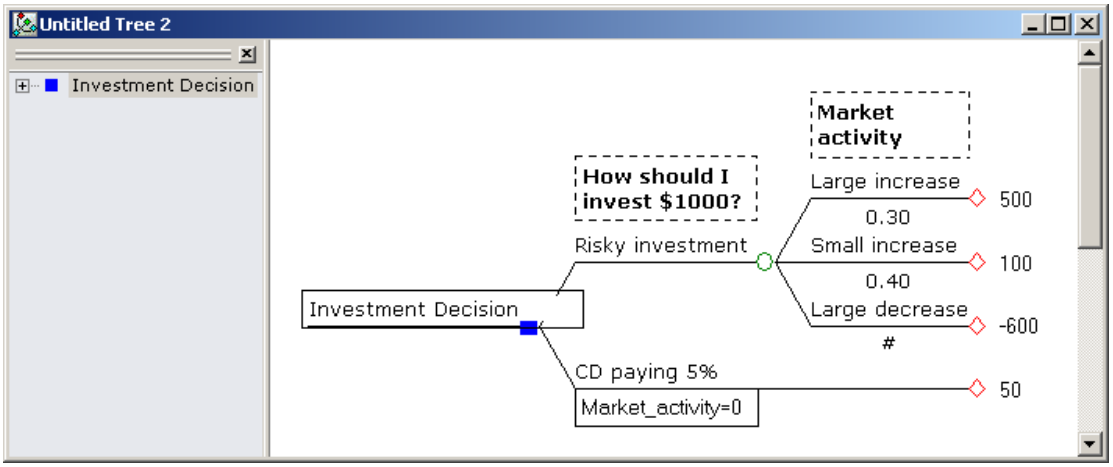
Converting and calculating the model

In TreeAge Pro, all analyses are performed on trees. To perform calculations on the model you have constructed, you must convert it into a tree. This will not affect your influence diagram, but instead will create a new tree document.

► **To convert an influence diagram to a tree:**

⇨ Choose **File** > Convert to Tree, or press the Convert tool bar button. 

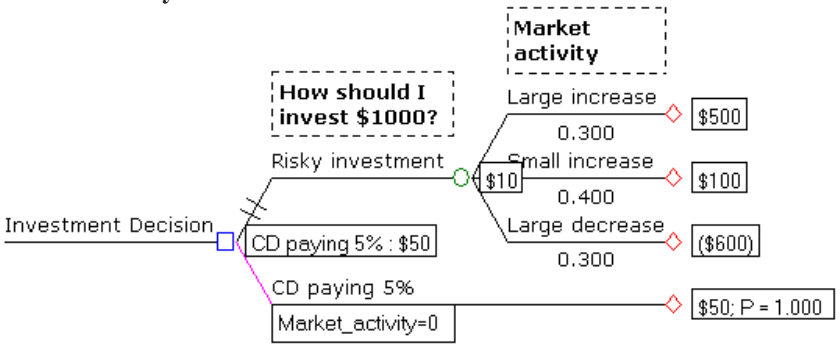
If no serious problems were found in your influence diagram, you will see the converted tree.



To see if the tree is complete, try calculating the model using roll back, described in Chapter 1.

► **To roll back the tree:**

⇨ Choose **Analysis** > Roll Back.



Notes on conversion

- IMPORTANT! There is no "hot-link" between a converted tree and the influence diagram used to create it. Changes made to the tree document will **not** be included in the influence diagram. To keep your influence diagram current, make structural and value changes in the influence diagram and convert it into a new tree when you need to check your work.
- Your converted tree will probably have some extra information initially — these are *node variables*. See Chapter 25 (and Chapter 12).
- Incomplete influence diagrams may still convert without error. For example, you may skip entering values and probabilities in the influence diagram to see the full tree structure.
- Failure to draw arcs between nodes where influences should exist, or the introduction of unnecessary arcs, can cause conversion problems.
- You are required to assign alternatives and outcomes to decision and chance nodes before conversion.

What's next?

This completes the basic influence diagram tutorial. You are now ready to use TreeAge Pro to build your own models.

Some questions you may have at this point:

- **How can I include my model in a presentation?** See Chapter 4, Printing and Presenting Trees and Influence Diagrams.
- **What if I have very complex cost formulas?** See Chapters 25 and 26, which cover a wide variety of influence diagram topics.

You might take some time now to review the topics in these chapters. As you work with the software and have questions about functionality, use this manual's table of contents or index, as well as TreeAge Pro's Help menu, to find answers.

To learn more about the use of influence diagrams, refer to one of the general decision analysis texts listed at the end of Chapter 1 (for example, *Decision Making and Forecasting*, or *Making Hard Decisions*).

Chapter 4

Printing and Presenting Trees and Influence Diagrams

This chapter provides basic instructions on how to customize printouts of your TreeAge Pro models, and how to import pictures of models into document and presentation programs like Microsoft® PowerPoint™, Excel™, or Word™.

In this chapter:

- Printing 42
- Exporting pictures 44
- Roll back (.TRB) files..... 47

Printing

Printing documents in TreeAge Pro is similar to printing documents from any other Windows application.

► **To print an active document without customization:**



▷ Choose **File >** Print..., and press ENTER or click OK.

A small model, like the Stock Tree or influence diagram, easily prints on a single page. A large tree may not initially print on a single page, but you may be able to get it to fit by shrinking it, or by changing the page orientation to landscape. As with other Windows applications, before printing you can select a different printer, or change the selected printer's settings by selecting File > Page Setup….

Adjusting printouts using print preview

In the Print Preview dialog, you can see how your printed document will appear (although fonts may be distorted). You can also adjust the size and location of the model on the printout.

► **To preview the print layout of a document:**

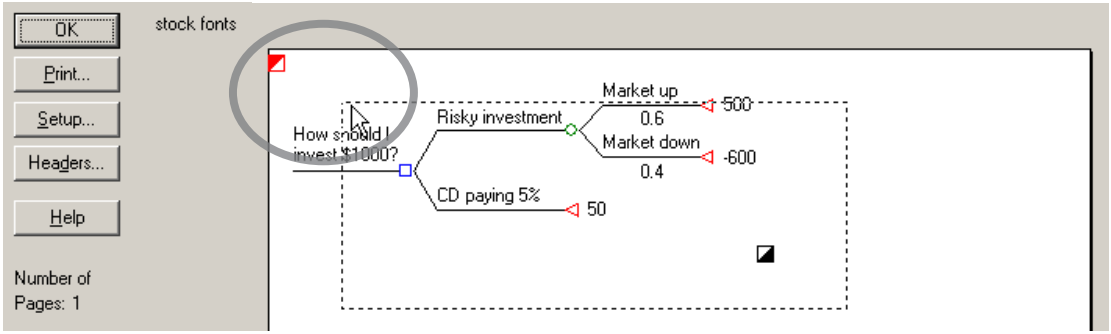


▷ Choose **File >** Print Preview....

The picture will be aligned to the upper left corner of the preview area. A half-filled red square should be visible at the top left of an invisible rectangle surrounding the picture, and a similar black square at the bottom right. These squares enable you to move and resize the picture within the printout.

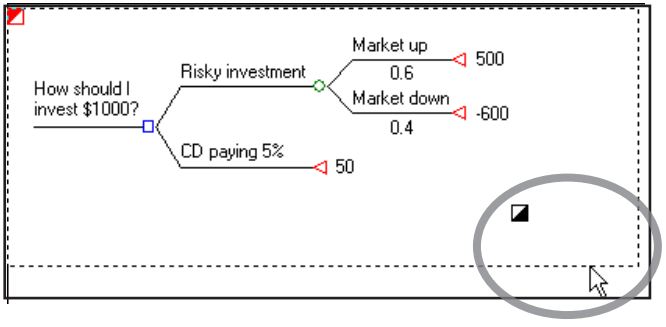
► **To move the picture within the printed page:**

▷ Click and drag the red square in the upper left corner of the picture.



► **To resize the picture within the page:**

▷ Click and drag the black square. The growing (or shrinking) picture-rectangle retains its proportions, to prevent distortion of the picture.



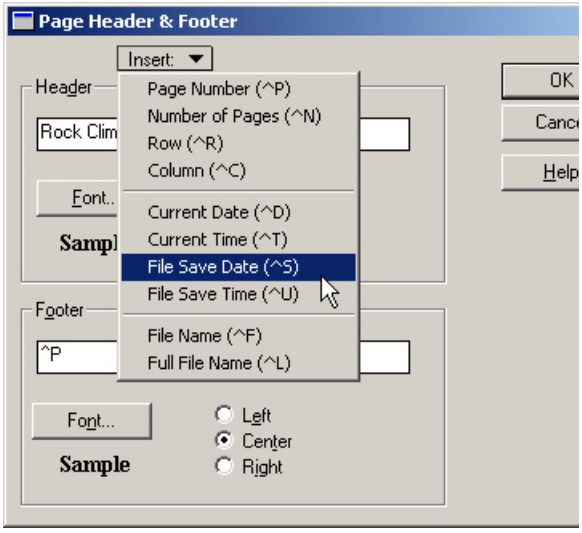
Headers and footers

Headers and footers can be printed with any document (including influence diagrams and graphs). Headers and footers can be set up in the Print Preview dialog.

► **To add headers or footers to a document:**

▷ Click on the **Headers...** button in the Print Preview dialog.

▷ Enter the desired text in the Header field and/or the Footer field. Add special text (page number, date, file name, etc.) using the Insert pop-up menu.



▷ Change the font and alignment of the text by clicking the appropriate buttons.

▷ Press ENTER or click OK to return to the Print Preview dialog.

Headers and footers that you setup will appear in the print preview.

Headers and footers can also be set up using the Preferences dialog, as described in Appendix C.

Printing preferences

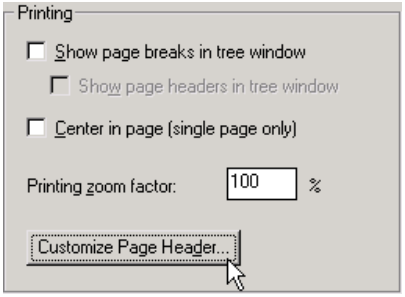
Several additional features for customizing printing are available in the Printing page of the Preferences dialog.

► **To change printing preferences:**

▷ Close the Print or Print Preview dialog, if open.

▷ Choose **Edit >** Preferences (or press F11).

▷ Select the Printing category.



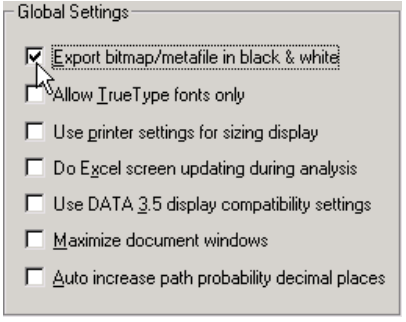
The headers and footers can also be customized from this dialog.

▷ Click OK to leave the Preferences dialog.

Details about printing preferences can be found in Appendix C, “Preferences.”

Global preferences

Some additional settings for customizing printing are available in the Display page of the Preferences dialog, under Global Settings.


Global Settings
Export bitmap/metafile in black & white
Allow TrueType fonts only
Use printer settings for sizing display
Do Excel screen updating during analysis
Use DATA 3.5 display compatibility settings
Maximize document windows
Auto increase path probability decimal places

Exporting pictures

To include a picture of a tree or other TreeAge Pro document in another program, such as Microsoft® Word™ or PowerPoint™, you can either export a picture to a graphics file, or copy it over the clipboard. Note the following differences between the two methods:

- To get a picture of only part of tree, use the copy method.
- Copying a picture of a tree or subtree over the clipboard will not capture arrows or unbound note boxes.
- Copying a picture of a tree or subtree over the clipboard will show all nodes as selected, using solid node coloring.
- A picture copied to the clipboard will be *pasted* into the target document. A picture exported to a graphics file will be *inserted* (or placed) into the target.

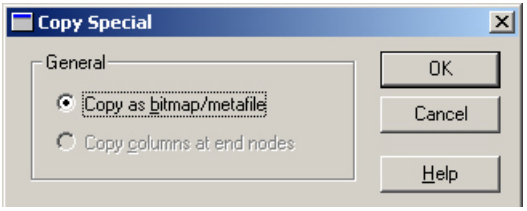
Consult the target document program's documentation for more information on pasting and then manipulating copied pictures. More information on choosing between available graphic formats is provided later in this chapter.

Copying a picture of an influence diagram or graph

In an influence diagram or graph window, it is not necessary to select anything before copying the picture.

► **To copy a picture of an influence diagram or graph to another program via the clipboard:**

▷ Choose **Edit >** Copy Special…, or > Copy as Bitmap/Metafile. If Copy Special... is used, select the Copy as Bitmap/Metafile in the ensuing dialog.



▷ Switch to the target document in the other program, and choose Edit > Paste, or > Paste Special.

Copying a picture of an tree

To copy a picture of a tree to the clipboard, you must first select it.

► **To copy an entire tree to the clipboard:**

▷ Hold down the CONTROL key and click on the root node. Or, select the node and choose **Options >** Select Subtree.

▷ Choose **Edit >** Copy Special…, or > Copy as Bitmap/Metafile.

It is also possible to copy only part of a tree.

► **To copy a partial "subtree" to the clipboard:**

▷ Hold down the CONTROL key and click on an internal node to select its subtree (or, choose Options > Select Subtree).

▷ Choose **Edit >** Copy Special… or > Copy as Bitmap/Metafile.

Subtrees — parts of trees — are explained in more detail in Chapters 8 and 9.

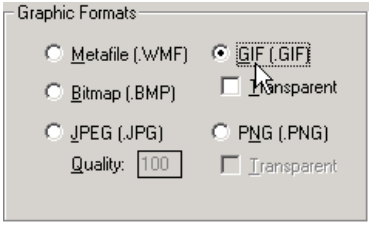
Exporting a picture to a file

Unlike copying a picture over the clipboard, exporting a graphics file includes all of a model's visual elements, including note boxes and arrows.

► **To export a picture of the active document (tree, influence diagram, or graph) to a grahics file:**

▷ Choose **File >** Export….

▷ In the Export Format dialog, pick the desired format for the saved graphics file (e.g., metafile, JPEG, bitmap).



▷ If you specify the JPEG format, you will need to specify the quality of the image — if file size is not an issue, specify the highest quality setting, 100.

▷ Click OK to display the standard "Save As" dialog. Note where you save the file, as you will need to find it in order to import it or open it in another program.

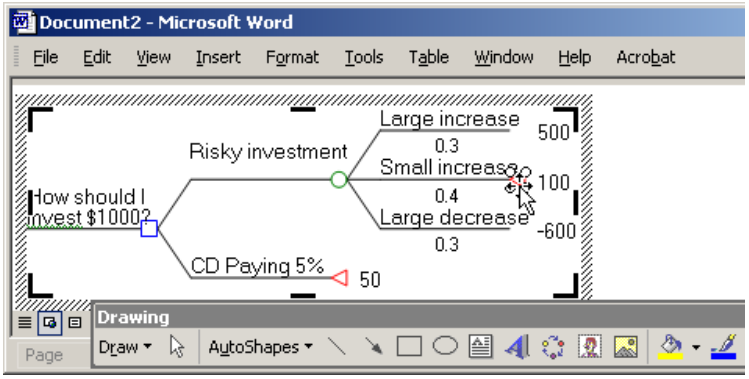
Working with exported graphics in other programs

To insert a saved graphic file to a Microsoft® Office™ document, use the appropriate program's Insert menu — usually Insert > Picture > From File. If you are using other presentation software, consult that program's documentation for instructions on inserting files.

Graphics design applications can be used to open and edit exported graphics, in order to edit the picture and save it in other formats. For additional information about working with the different graphic file formats in other programs, see the next section.

The graphics export formats supported by TreeAge Pro are described here. Added in v2006 is the PNG format.

Metafile (.wmf files) – A Windows *metafile* is a lot like a document. It stores text, shapes, and their exact layout, which are then reproduced on-screen in your word processing or presentation program, and then on your printer. A metafile is a much smaller file than a bitmap, but will always print at the highest resolution available — just like other documents. Metafiles inserted into Microsoft® Office™ documents, or opened in a graphics programs like Paint Shop Pro™ (Jasc Software), can be edited. The text, colors, and other elements of the picture can be modified relatively easily, using drawing and text tools. The image can also shrink or stretch without losing quality.



Bitmap (.bmp file) – In most cases, PNG or JPEG formats will be preferred over BMP files. BMP graphics are very large file, even for moderately-sized trees. See next section.

Portable Network Graphics (.png file), GIF (.gif file), and JPEG (.jpg file)

Increasingly, the PNG format is preferred for program screenshots over bitmaps and JPEGs because of its small file size, transparent background option, and equivalent quality. A JPEG is a "lossy" compressed bitmap, whereas a PNG is "losslessly" compressed.

With JPEGs, a high quality setting (higher number) should be specified when you export the file, in order to get the same quality as a PNG or standard bitmap. Exported bitmaps (including PNG and JPEGs) do not always resize smoothly, their content cannot be modified, and they do not print at high resolution. They are the standard for on-screen presentation, or in web pages.

Both PNG and GIF graphics have a transparency option.

Using a screenshot capture program

In cases where none of the available formats are appropriate for your exported picture of a TreeAge Pro model or graph, a third-party screenshot capture program may be used. Or, simply use the ALT+Print Screen keyboard combination, and paste the result into a word processing document or email.

Manipulating images using TreeAge Pro Interactive library or Excel module's TreeAgeProLib

TreeAge Pro Interactive is a programming interface to TreeAge Pro models, which can be used to create scripts in other programs (like Excel macros) to open and analyze models built in TreeAge Pro, and also export PNG and GIF graphics of models and graphs.

See Appendix F for information on using TreeAge Pro Interactive or the Excel module's TreeAgeProLib.

Roll Back (.TRB) files

A roll back file is an uneditable copy of your rolled-back tree. You may use this type of file to distribute a viewable representation of your calculated tree to other TreeAge Pro users. The file type is proprietary; .TRB files can only be opened using TreeAge Pro.

When a .TRB file is opened, the tree is already rolled back. The values and structure of the tree are fixed, but display settings can be changed.

► **To create a roll back (.TRB) file:**

▷ Roll back your tree.

▷ While the tree is still rolled back, choose **File >** Export.

▷ In the Export Format dialog, select the button named Rollback Format (.TRB).

▷ Save your .TRB file using the standard Save As dialog box which appears.

Working with Decision Trees

The chapters in Part II provide provide information about commonly-used tree-building and analysis features in TreeAge Pro 2007.

In this part:

- Chapter 5, "Analyzing the Tree"
- Chapter 6, "Graph Windows"
- Chapter 7, "Tree Calculation Methods and Preferences"
- Chapter 8, "Selecting Subtrees and Multiple Nodes"
- Chapter 9, "Making Changes to Tree Structure"
- Chapter 10, "Annotating the Tree"
- Chapter 11, "Customizing the Tree Display"
- Chapter 12, "More Tree-Building Tools"

Chapter 5

Analyzing Decision Trees

This chapter describes in detail the variety of expected value analyses available in TreeAge Pro.

Sensitivity analysis and Monte Carlo simulation are covered in Part III of the manual. Cost-effectiveness and Markov analysis are covered in Part V.

In this chapter:

- Numeric formatting 52
- Expected values 53
- Roll back 54
- Rankings 55
- Standard deviation 56
- Probability distributions 57
- Expected value of perfect information 65
- Other analyses 69

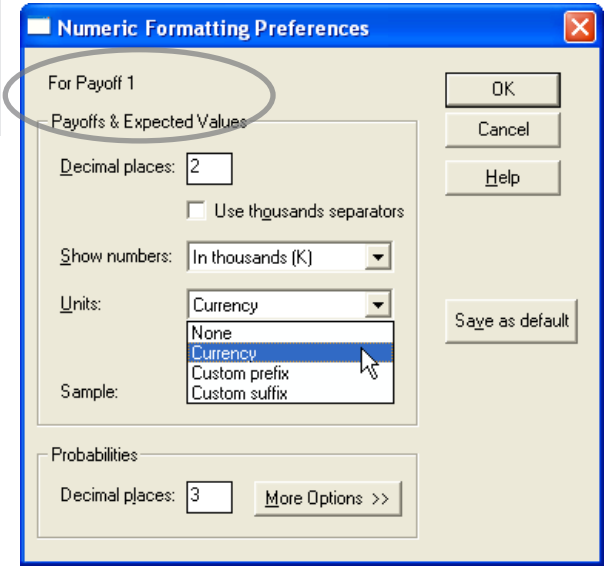
Numeric formatting

TreeAge Pro performs calculations at the highest available precision, but values are *displayed* using a specified number of decimal places — up to 9 — with the option to use abbreviations and unit symbols. As described at the end of Chapter 2, each tree and graph has its own set of numeric formatting preferences.

► **To view/modify a document's numeric formatting:**

▷ Choose **Edit >** Numeric Formatting, or press the F10 key. (Or, click the Numeric Format… button in the Calculation Method preferences category.)

In a tree, the numeric formatting dialog shows the label "For Payoff *X*," circled in the above picture, indicating the currently active payoff (#1–9, as specified in the Calculation Method preferences).



Under Payoffs & Expected Values, you can specify:

- **Decimal places**: The number of digits (0-9) after the decimal point to display for calculated values;
- **Use thousands separators**: Whether to incorporate thousands separators into numbers greater than 1000;
- **Show numbers**: Whether to show numbers exactly or abbreviated (K=thousands / M=millions / B=billions);
- **Units**: Whether to display numbers with a built-in or custom prefix or suffix.

If you choose "Currency" units, expected values will be reported using the type of currency set in your operating system's control panel. If you select either "Custom prefix" or "Custom suffix," you can enter units of your choice in the Tag editor.

Under Probabilities, you can specify the number of decimal places (0-9) to use when displaying calculated probabilities. (In some reports, very small values may display using scientific notation.)

Notes on entering numbers

- If your computer is set to use commas (",") to represent decimals, rather than periods (".") enter numbers in TreeAge Pro in this fashion, just as you would in a spreadsheet or calculator. Particular trees, however, can be set to override the computer's regional settings and reverse the usage of decimals; see Chapters 2 and 12 for details.
- Independent of numeric formatting preferences, payoffs and other values can be entered using K/M/B abbreviations and thousands separators (e.g., typing 2K is equivalent to typing 2,000). Scientific notation can also be used when entering very small or very large numbers (e.g., typing 1e3 is equivalent to typing 1000).

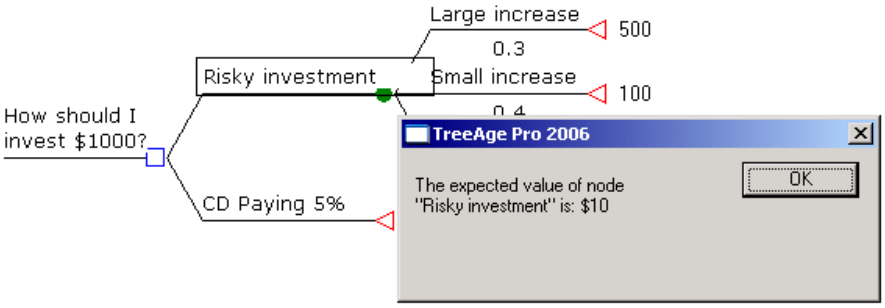
Expected values

In addition to calculating and displaying expected values for *all* nodes in a tree (see Roll back, below), TreeAge Pro can also report a single expected value for a selected node. This provides a useful method of verifying the completeness of a single part of an incomplete tree.

► **To calculate the expected value of a node:**

▷ Select a node.

▷ Choose **Analysis >** Expected Value.

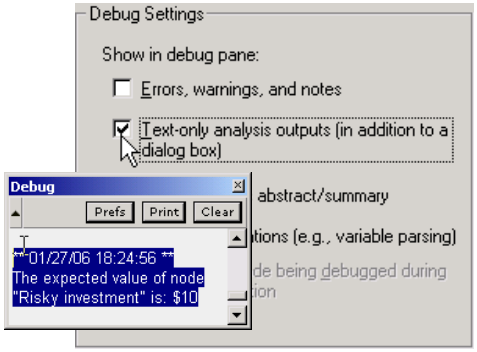


Using the Expected Value command with multiple nodes selected will report the sum of the nodes' expected values.

The result is displayed using the tree's current numeric formatting preferences; see the previous section, on numeric formatting, for details.

Using the expected value result

- Chapter 1, "Decision Analysis Primer," includes a detailed description of the basic concepts used in calculating expected values in decision trees. Refer to that chapter, or one of the books listed at the end of Chapter 1, for a review of basic concepts.
- The *debug pane*, which floats at the bottom of the window, can be used to copy the results from the expected value analysis and other tree calculations. Press F4 to show the debug pane, and click its Prefs button to turn on the output of text analysis results.



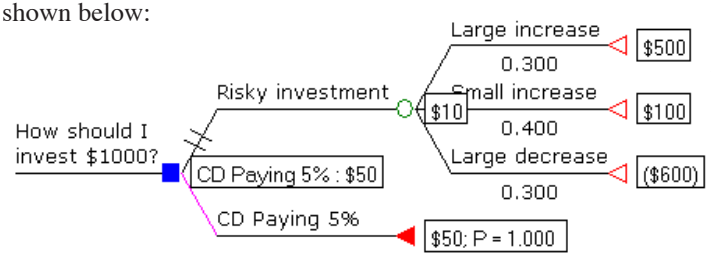
Roll back

As described in Chapter 1, roll back refers to the calculation of expected values starting at the terminal nodes and continuing back to the root node. In TreeAge Pro, a variety of information is reported in the tree when it is rolled back.

► **To roll back the tree:**

▷ Select **Analysis >** Roll Back (or click the beach ball icon in the tool bar).

The rolled back Stock Tree, from Chapter 2, is shown below:



Some tree commands are available only if roll back is turned off.

► **To turn off roll back:**

▷ Select **Analysis >** Roll Back again.

Roll back display details

- **Decision nodes:** A box to the right of the node reports the name and expected value of the preferred alternative. TreeAge Pro marks the branches of non-optimal alternatives using hashes, and colors the optimal branch.
- **Chance nodes:** A box to the right of the node reports the expected value. Probabilities are calculated (if necessary) and displayed beneath the node's branches.
- **Terminal nodes:** A box to the right of the node reports the payoff. If the node is in the optimal path (i.e., is an outcome of the preferred strategy), its *path probability* ("P =…") is also reported.

Customizing the roll back display

A picture of the rolled-back tree can be printed or exported to a graphic file, as described in Chapter 4.

There are many ways to customize the appearance of the rolled-back tree prior to printing or exporting it. In addition to the features described below, Chapter 11 covers a number of important tree display preferences related to roll back (for example, setting up *terminal node columns*).

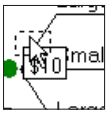
Values displayed in the rolled back are displayed using the tree's numeric formatting settings; see the beginning of this chapter for details on modifying these preferences.

When roll back is turned on, initially all nodes in the optimal path of the are selected (highlighted). Clicking on the tree will deselect these nodes.

Occasionally, a roll back box will cover the text of a branch description or probability. This can be corrected by moving the box.

► **To move a roll back box:**

▷ Click and drag the box to a better location.



It is also possible to hide individual roll back boxes.

► **To hide a selected node's roll back box:**

▷ With the tree rolled back, right-click on the node whose roll back box you want to hide.

▷ From the pop-up quick menu, choose Hide roll back box.

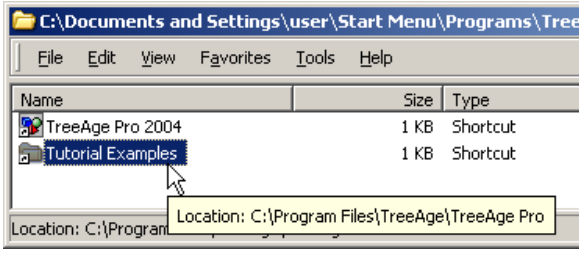
Roll back box notes:

- Roll back box state will be saved with the tree.
- A hidden roll back box can be redisplayed by right-clicking on the node and choosing Hide roll back box again.
- If you want to reset hidden and moved roll back boxes in a section of the tree, cut the subtree and then paste it back into place.

Rankings

The Rankings analysis, available when a single decision node is selected, displays a text report listing the alternatives at that node and their expected values, in rank order.

Try the Rankings analysis on the example tree called Rankings Analysis, which is a slightly more complex version of the Stock Tree from Chapter 2. A copy of the tree can be found in TreeAge Pro's Tutorial Example subdirectory. The Start menu includes a shortcut to this subdirectory.


C:\Documents and Settings\user\Start Menu\Programs\Tree
File Edit View Favorites Tools Help
Name Size Type
TreeAge Pro 2004 1 KB Shortcut
Tutorial Examples 1 KB Shortcut
Location: C:\Program Files\TreeAge\TreeAge Pro

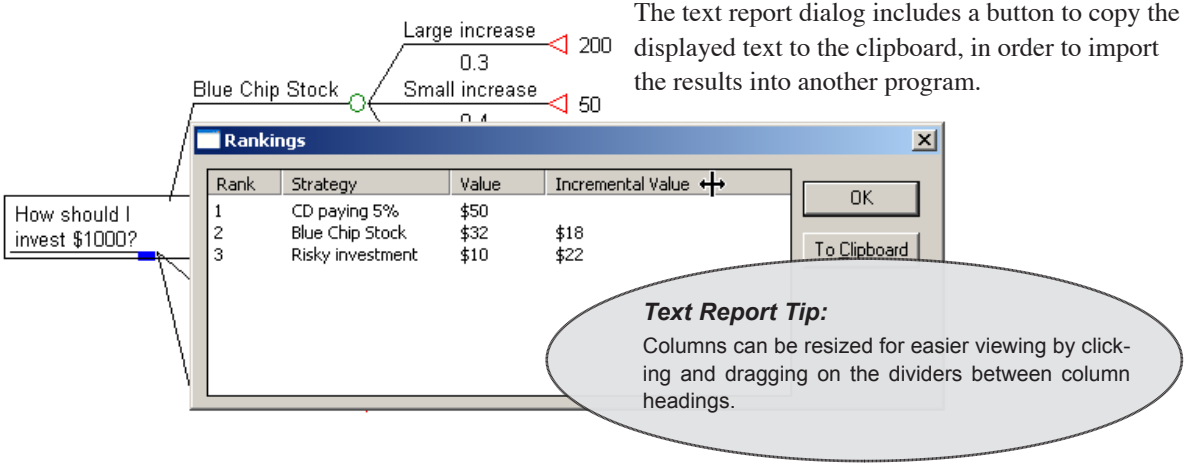
► **To calculate and rank decision alternatives:**

▷ Select the decision node.

▷ Choose **Analysis >** Rankings.

A text report appears which ranks the options, and specifies their expected values. In the case of sub-optimal options, it also specifies a marginal (or *incremental*) value — the amount by which one option is outperformed by the next best option.

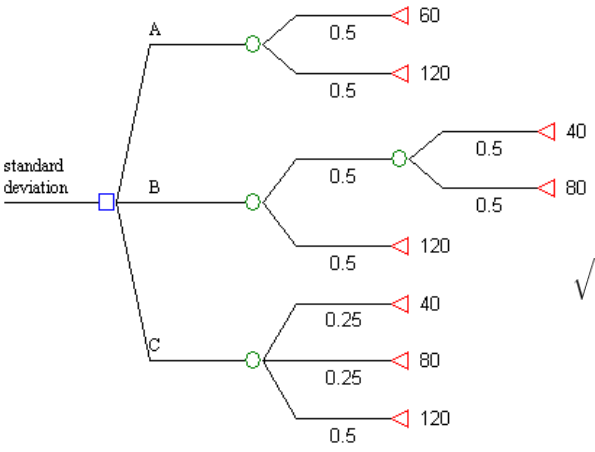
The text report dialog includes a button to copy the displayed text to the clipboard, in order to import the results into another program.



Standard deviation

In addition to comparing strategies based on their expected values, TreeAge Pro also offers several ways to look at an option's *risk* — the degree of variability in outcomes. The basic, statistical measure of risk is *standard deviation*. In TreeAge Pro, a standard deviation can be calculated for a single strategy (or any other chance node), based on the path probabilities and payoffs of all terminal nodes in its path.

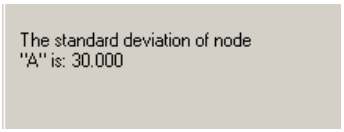
To try the standard deviation calculation, use the example decision tree called Standard Deviation (in TreeAge Pro's Tutorial Examples subdirectory). In this tree, despite the fact that the three alternatives look different, a rankings or roll back analysis is *indifferent* between them — all three have the same expected value, 90. In this case, a choice might be based on minimizing risk as measured by standard deviation.



► **To calculate a standard deviation:**

▷ Select the chance node labeled *A*.

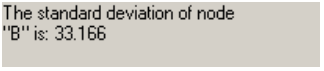
▷ Choose **Analysis >** Standard Deviation.

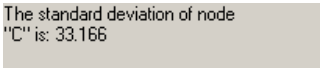


The calculation used is:

$$\text{SD}(\textit{Strategy A}) = \sqrt{0.5(60-90)^2 + 0.5(120-90)^2} = 30$$

where 90 is the strategy A's expected value, or mean. Compare this to the calculated standard deviations for strategies B and C:





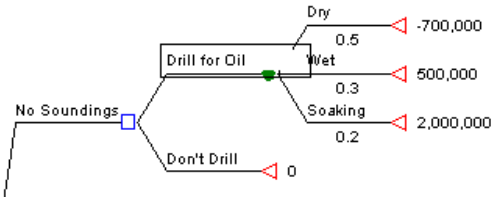
If you were to choose a strategy based on minimizing risk, as measured by standard deviation, strategy A would be preferred. Note that B and C are statistically identical, having the same terminal node values and path probabilities, and thus the same standard deviation:

$$\sqrt{0.25(40-90)^2 + 0.25(80-90)^2 + 0.5(120-90)^2} = 33.166$$

Probability distributions

The risk associated with alternatives under consideration can be displayed graphically, using a probability distribution histogram, or *risk profile*. A probability distribution graphs the values (i.e., payoffs) and path probabilities of all terminal nodes within a strategy.

Open a tree called Oil Drilling Problem, found in TreeAge Pro's Tutorial Examples subdirectory. This model has some interesting elements, including multiple decisions nodes.



To create your first probability distributions using TreeAge Pro, start by analyzing a chance node.

► **To view a probability distribution histogram:**

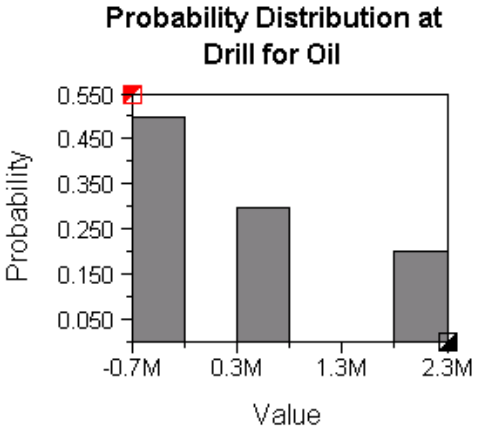
▷ Select a chance node – in this case the topmost chance node labeled Drill for Oil, in the No Soundings section of the tree.

▷ Choose **Analysis >** Probability Distribution. 

TreeAge Pro displays the analysis results in a graph window.

Graph window contents

Graphs created in TreeAge Pro are documents that can be printed, saved, or exported to a graphic file (or an Excel chart, as described in Chapter 28). See Chapter 6 for information on customizing a graph's text labels, line markers, axes, and other visual elements.



Every graph window includes a row of buttons at the top. The **Actions** button includes the same commands found in the **Graph** menu. Additional buttons and controls may be added for special functions, or commonly-used commands. The probability distribution graph includes three buttons.



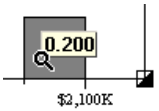
- **Options** – Opens the Graph Options dialog, which has multiple categories. Options specific to probability distributions are covered in this chapter; see Chapter 6 for information on the axes and lines categories.
- **Excel Chart** – See note on next page.
- **Actions –** Various graph commands, including: **Text Report…** for viewing and exporting of the calculated values underlying the graph; **Distribution Statistics…** for reporting the mean, standard deviation, and other basic summary statistics of the distribution; see Chapter 6 for information on the other Actions/Graph menu commands.

Using the probability distribution graph

In the example histogram shown on the previous page, TreeAge Pro displays a separate bar for each possible payoff. However, this will not usually be the case, as a reasonable number of bars will be used to represent the probability distribution.

The value range is divided into a reasonable number of intervals, initially. The height (i.e., probability) of a bar is the sum of the path probabilities of all terminal nodes with payoffs in that interval. The height of all bars sums to 1.0 (100%). The vertical axis will scale to the height of the highest bar.

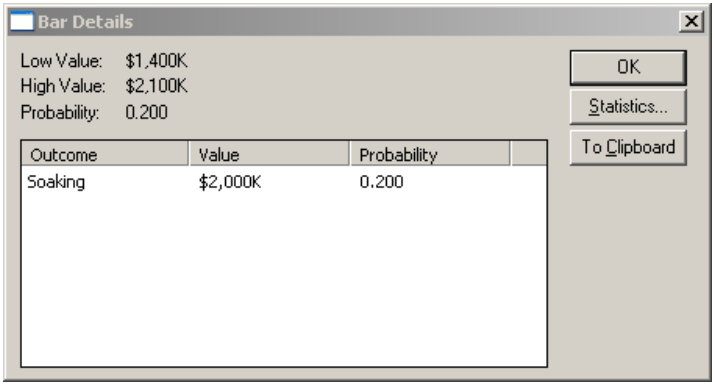
Clicking once on a histogram bar will display the total probability for that interval.



To get a detailed breakdown of the terminal nodes summarized by the bar, use the Bar Details text report.

► **To see a report for all outcomes included in one bar:**

▷ Double-click on a bar in the histogram.

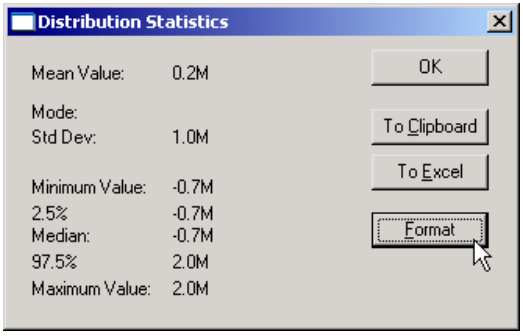


▷ To copy the report to the clipboard, in order to paste it into a spreadsheet or other document, click To Clipboard.

▷ Click OK to return to the graph window.

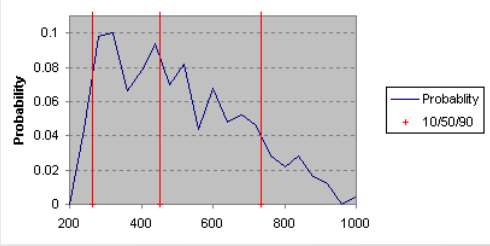
► **To export statistics for the analysis**

▷ Choose **Graph** > or **Actions** > Distribution Statistics…, and then copy the report to the clipboard



Excel module users:

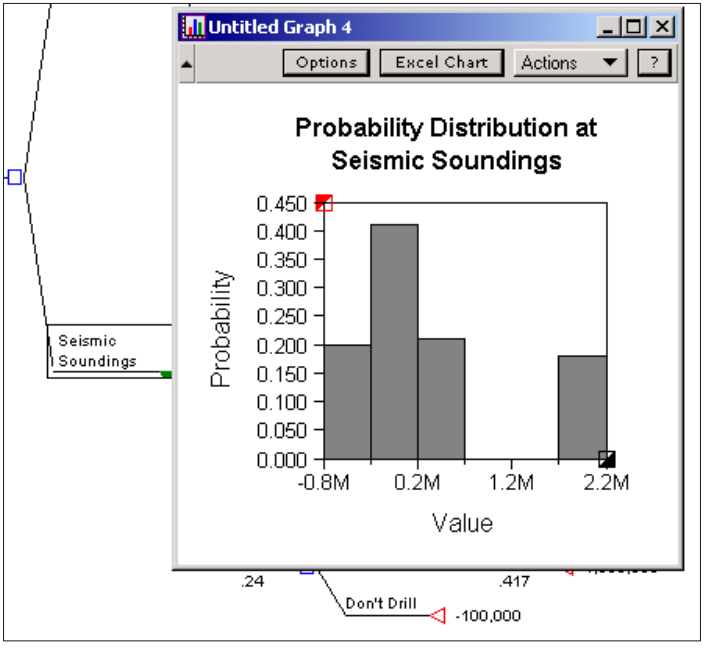
A distribution graph can be displayed in TreeAge Pro as either a non-cumulative histogram (bar chart), or cumulative line graph (see below). With the optional Excel module, a distribution graph exported as an Excel chart can be changed to a non-cumulative line graph (see the example at left). The Excel chart also adds lines marking the 10/50/90 percentiles. See Chapter 28 for information on Excel charting.



Downstream decision nodes

As noted above, the Oil Drilling Problem tree includes multiple decisions. To see how the probability distribution analysis is affected by downstream decision nodes, analyze the leftmost chance node in the tree, found in the Seismic Soundings *subtree* (i.e., portion of the tree). Each of the Soundings node's branches is a decision node.

If there are decision nodes anywhere to the right of the analyzed node, as in this analysis, TreeAge Pro calculates expected values, and then selects an optimal strategy at each downstream, or *deferred*, decision. The histogram only includes terminal nodes from the optimal path, therefore the path probabilities in the histogram will still sum to 1.0. To see which terminal nodes are in the optimal path, you can roll back the tree.



Outcome	Value	Probability
Don't Drill	($100K)	0.410
Wet	$400K	0.120
Wet	$400K	0.090

Double-clicking on the middle bar will show the three outcomes whose payoffs fall into that interval, and whose path probabilities add up to the bar's height.

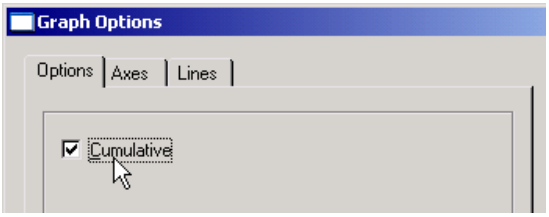
Cumulative probability distributions

A probability distribution bar graph can also be displayed in cumulative form. Rather than using discrete bars to indicate the probability of outcomes *within* specific payoff intervals, the cumulative graph shows a continuous series of bars. The top of each interval's bar indicates the probability of an outcome within or less than the interval's top value.

Try the cumulative setting on the probability distribution graph generated on the previous page, in the Soundings tree.

► **To change a probability distribution to cumulative form:**

▷ In the graph window, click on the **Options** button (or choose Graph > Options…).

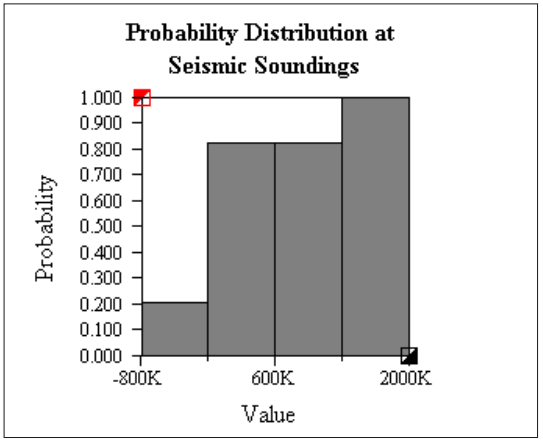


▷ Under the Options category, check the Cumulative option. Click OK to return to the Graph window.

TreeAge Pro does not generate a new graph window, but instead changes the current graph to the cumulative form.

► **To revert back to the non-cumulative form of histogram:**

▷ Re-open the Graph Options dialog, and uncheck the Cumulative option.



Comparative probability distributions

Multiple, cumulative probability distributions can be displayed in a single graph, allowing graphical comparison of options. In a comparative distributions graph, the cumulative distributions are displayed in outline, instead of using filled bars. This format enables graphical comparison of strategies' based on their risk profile.

Details about the graphical interpretation of comparative probability distributions is provided later in this chapter, in the section on dominance.

When a single decision node is selected, the Analysis menu displays the Comparative Distributions command. The comparative distributions analysis can be tried at the root, decision node in the Oil Drilling Problem tree.

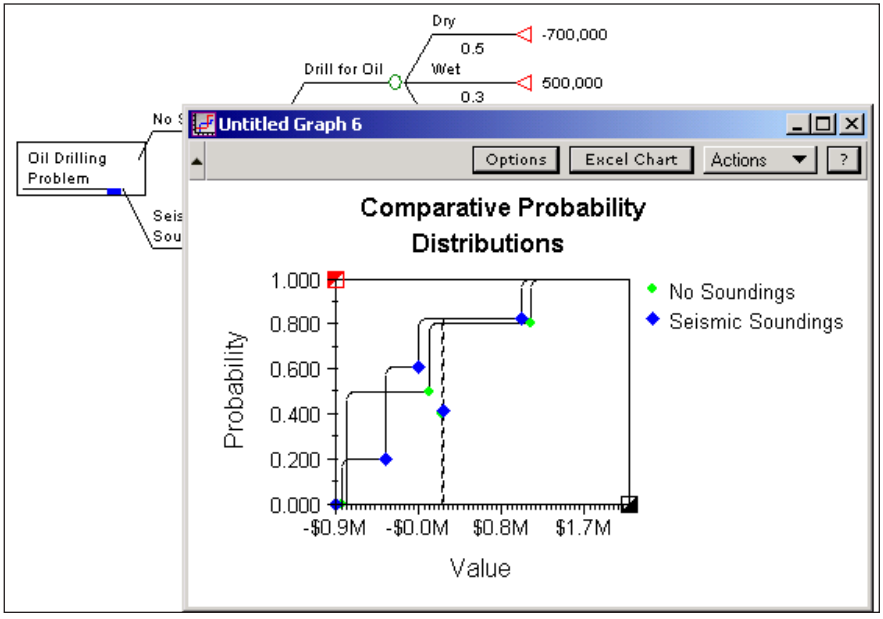
► **To generate a comparative distribution line graph:**

▷ Select the root, decision node.

▷ Choose **Analysis >** Comparative Distributions.

The resulting graph displays the outlines of the cumulative probability distributions for the competing options Seismic Soundings and No Soundings. The outline for a strategy is marked at each corner (where the cumulative probability "curve" rises) with that strategy's symbol, as listed in the legend to the right of the graph.

The expected value of a strategy is indicated in the graph using a vertical, dotted line labeled with the strategy's legend symbol. Here, Seismic Soundings has the higher expected value by a small margin.



There are two options for displaying the lines in the comparative distribution graph. In addition to the step-wise outlines initially used in the graph, shown above, lines can also be drawn that directly connect the top-left corner of each interval's bar.

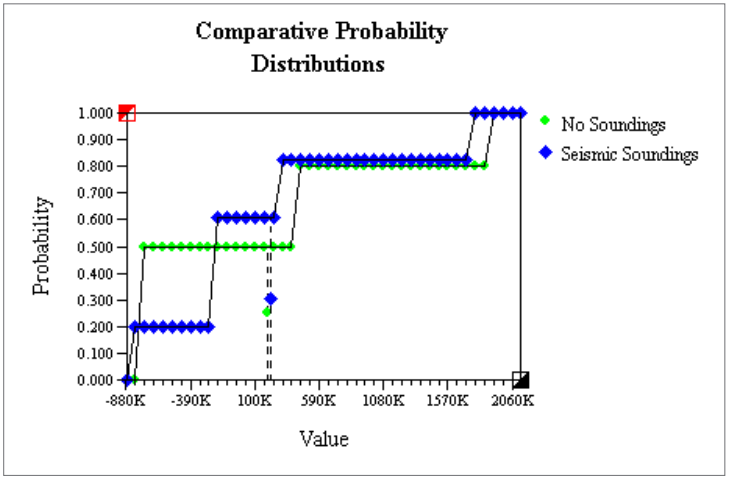
► **To change the comparative distribution's line format:**

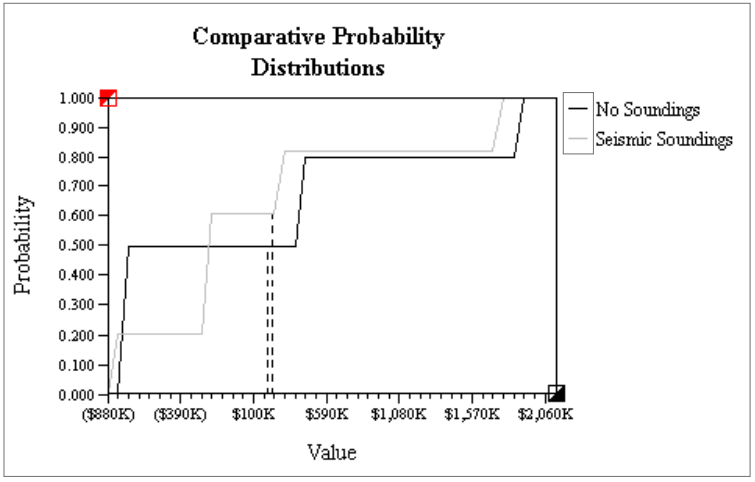
▷ In the graph window, click on the **Options** button.

▷ Under the Options category, check the Draw lines directly… option. Click OK to return to the Graph window.

Using numerous, short intervals helps to differentiate strategies in the comparative distributions graph. However, the many line markers that appear in the direct-line version of the graph may be distracting.

Double-clicking on a line marker in the legend opens the Modify Line Marker dialog, which can be used to turn off line markers and to specify that colored lines be used instead, as in the example below. Details on modifying graph lines and line markers are given in Chapter 6.





Dominance in probability distributions

Comparative probability distributions can be interpreted graphically, by evaluating conditions of *dominance*. There are two types of dominance that can be identified relatively easily: deterministic and stochastic (also called absolute and extended dominance). Conditions of dominance can provide more insight into a decision than simple expected value comparison.

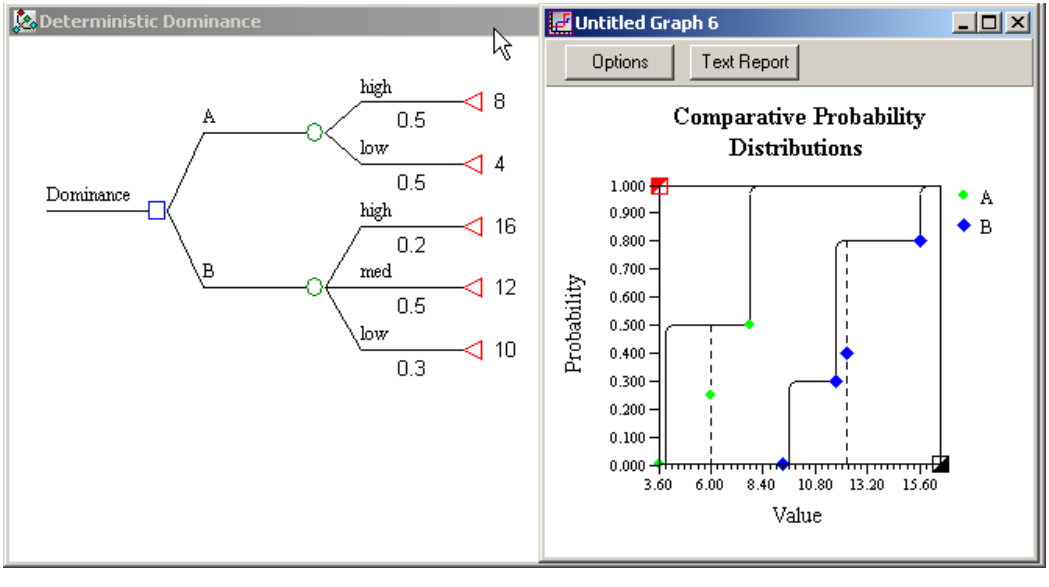
Deterministic dominance occurs when one option not only has the best expected value, but its worst possible outcome is better than (or equal to) the best outcome of any other option. It can be identified as follows:

- if optimization requires *maximizing* value (e.g., profit), the worst "bar" of the dominant option (its left-most vertical line) lies on, or to the right of, the best (right-most) "bar" of the dominated option(s);
- if optimization requires *minimizing* value (e.g., costs), the worst "bar" of the dominant option (its right-most vertical line) lies on, or to the left of, the best (left-most) "bar" of the dominated option(s).

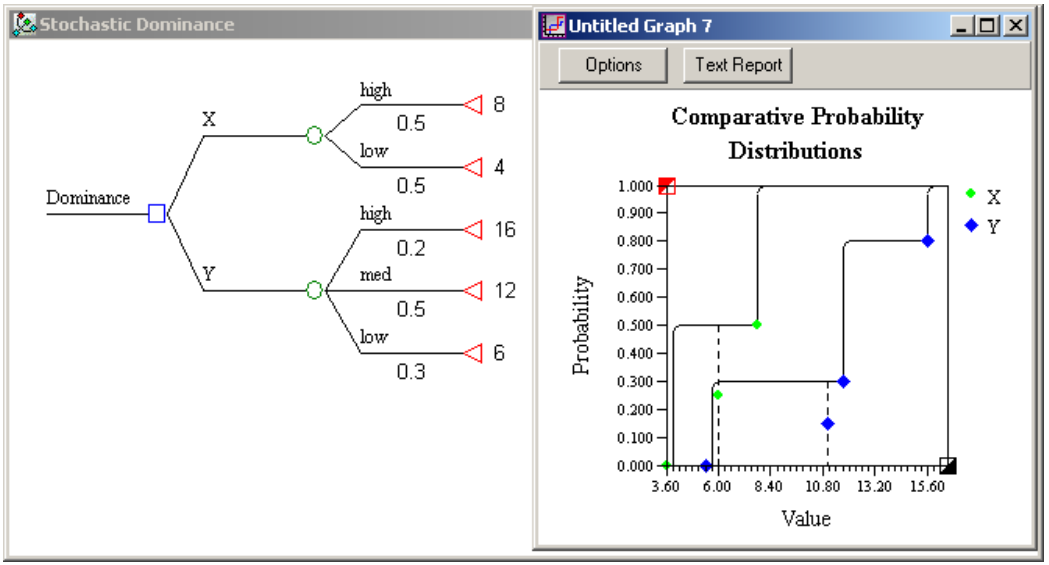
Both situations can be visualized in the graph below; note the gap between A's highest value, 8, and B's lowest value, 10. Assuming expected values are being maximized, B dominates A. If, instead, expected values are being minimized, then A dominates B.

The kinds of problems in which decision analysis is applied will not often display deterministic dominance, however. *Stochastic* dominance is more likely. Conditions of stochastic dominance — also called extended or probabilistic dominance — are identified as follows:

- if optimization requires *maximizing* value, the entire outline describing the stochastically dominant option lies to the right of the dominated option's outline — the lines can touch for part of the graph, but never cross;
- if optimization requires *minimizing* value, the outline describing the stochastically dominant option lies on, or to the left of, the dominated option's outline.



Just as in looking for deterministic dominance, the critical section of the graph in the case of stochastic dominance is found between the lowest value of the dominant option and the highest value of the other. In this case, instead of a gap on the x-axis, there is overlap. This is illustrated in the graph below.



Since the lines do not cross, some kind of dominance must exist. X's highest value is greater than Y's lowest value, so it a condition of stochastic, not deterministic, dominance. Assuming that the objective is to maximize profit, Y is the stochastically dominant option.

If neither form of dominance exists, the strategies' lines will cross at one or more points.

Expected value of perfect information

Assume that you could buy information that perfectly predicted the outcome of a future uncertainty. What would this information be worth to you?

In a decision tree, the option to acquire perfect information (about a single uncertainty) can be modeled by moving the chance node representing the uncertainty to the left of a decision.

Although the uncertain event still follows the decision in time, the decision maker is assumed to have a perfect predictor of the event outcome before making the decision. Keep in mind that perfect information does not mean that you can *control* the event's outcome, only that you can *predict* the outcome.

Ignoring for the moment the cost of the perfect information, the revised tree cannot have a worse expected value than the original tree, and may have a better expected value. This difference in expected value is referred to as the expected value of perfect information (EVPI).

While predictive information is rarely perfect, the usefulness of EVPI is in calculating a maximum reasonable price for information. If perfect information in a particular situation has a base value of *x*, one should certainly not pay more than *x* for *imperfect* information. To see how imperfect information is dealt with in decision analysis, see Chapter 23, Bayes' Revision.

Notes on Value of Information analysis:

- For information on a somewhat more flexible implementation of EVPI using influence diagrams, see Chapter 25.
- Monte Carlo simulation in TreeAge Pro can be used to calculate expected value of perfect information for any number of predictable or resolvable uncertainties; see Chapter 16.

How EVPI is calculated

Before trying TreeAge Pro's shortcut for calculating EVPI in a decision tree, it is instructive to work through the extra steps required to calculate EVPI manually.

▷ Open the file Stock Tree from TreeAge Pro's Tutorial Examples subdirectory.

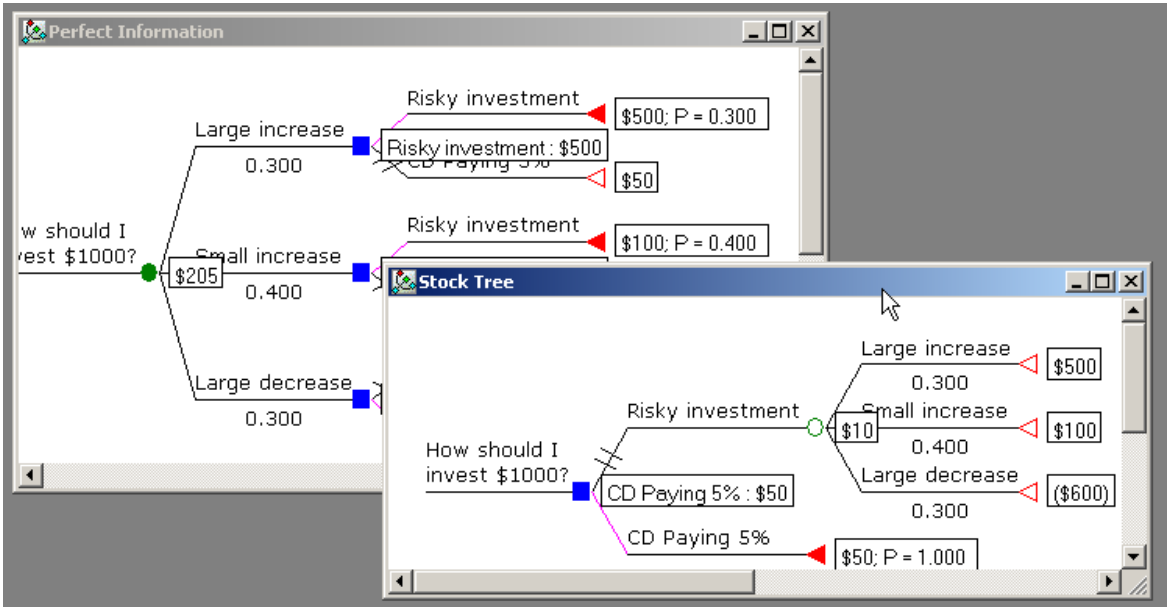
▷ Also open the example file called Perfect Information from the Tutorial Examples subdirectory. This version of the investment problem shows the time reversal of the Market uncertainty and the decision.

The two trees are shown on the following page.

▷ Roll back Stock Tree. The tree's expected value is $50 (equal to CD paying 5%'s value).

▷ Roll back the Perfect Information tree. The root node's expected value is $205.

▷ To calculate EVPI, take the difference between the expected value of the Stock Tree and that of the Perfect Information tree. The difference is $155. (If you were minimizing costs rather than maximizing profit, you would subtract the perfect information value from the regular expected value.)



In the investment model, the expected value of having perfect information about the market activity is $155. This is the most you should be willing to pay to obtain this information, and it affords some basis for appraising the value to you of a less than perfect predictor of market activity.

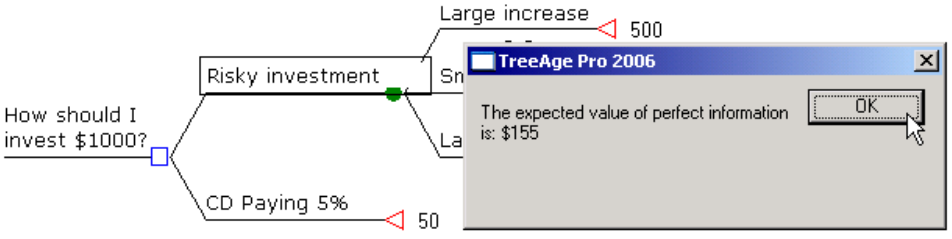
Now, try TreeAge Pro's shortcut for calculating EVPI in a tree. It requires only the original Stock Tree.

► **To calculate EVPI automatically:**

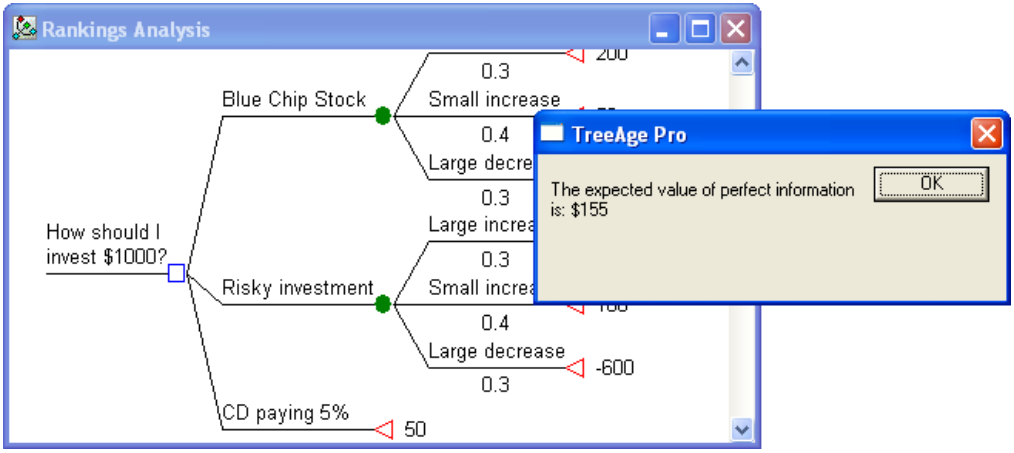
▷ Open the Stock Tree.

▷ Select the Risky investment chance node and choose **Analysis >** Expected Value of Perfect Info.

A dialog reports the value of $155.



It is also possible to calculate EVPI automatically in a decision tree when the same event appears in more than one strategy, as in example shown below. The assumption in the model is that each of the two stock investments under consideration is followed by the same uncertainty — whether the market will be up or down at the end of the year.



In order to calculate EVPI in this model, both market uncertainty chance nodes must be selected.

► **To calculate EVPI for the same chance node in multiple paths:**

▷ Before choosing Analysis > Expected Value of Information, *select all nodes in the tree which represent the same event*.

If multiple chance nodes are analyzed, they:

- must be descendants of the same decision node;
- must be "siblings"; and
- must have identical branches using identical probabilities. It does not matter if there are differences in the subtrees further to the right.

If there is more than one decision prior to the selected chance event, TreeAge Pro will prompt you to identify the decision for which EVPI should be calculated.

Avoiding EVPI errors

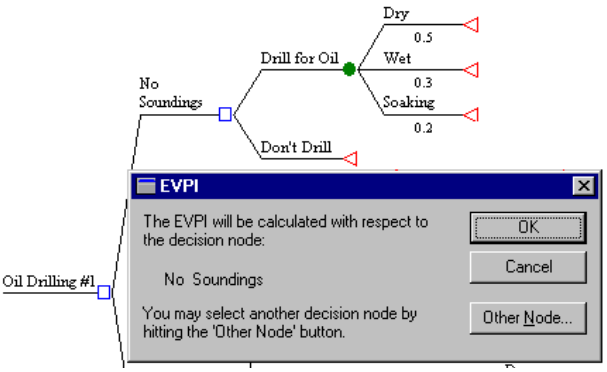
It is important to understand that it is possible to force invalid EVPI calculations. For example,

- ▷ Open the Oil Drilling Problem tree again.
- ▷ Select the Drill for Oil node in the No Soundings subtree and choose **Analysis >** Expected Value of Perfect Info.

In the resulting dialog boxes, you are presented with the option of having the analysis performed at the root, soundings decision node or at the drilling decision.

Performing the calculation at the No Soundings node is similar to the analysis undertaken above in connection with the EVPI tree. It certainly makes sense to calculate the value of knowing the state of oil reserves before deciding whether or not to drill.

However, what if you perform the EVPI calculation at the root decision node? The value reported is $437,500, or $87,500 higher than at the No Soundings node. Is this a meaningful EVPI calculation?



The structure of the tree *already* includes the option of securing *imperfect* information in the form of a seismic test — this is the initial decision. Performing EVPI by placing the chance node representing the uncertain amount of oil to the left of this decision is meaningless. Having already received *perfect* information, the decision whether to obtain additional *imperfect* information regarding the same subject should have no value or relevance. See Chapter 23, Bayes' Revision, for more information on imperfect information

Other analyses

The last section of this chapter covers three simple analysis tools.

This chapter does not cover all analyses. Refer to later sections of the user's manual for information on additional analysis options:

- Sensitivity analysis using variables is covered in Chapters 12 and 15. This includes 1-, 2-, and 3-way sensitivity analysis, as well as tornado diagrams.
- Monte Carlo simulation using distributions is covered in Chapters 16 and 17.
- Markov cohort analysis and microsimulation are covered in Chapters 34 and 36.
- Cost-effectiveness analysis is covered in Chapters 31 and 32.

Range of possible payoffs

This analysis will tell you the highest and lowest payoffs which may occur from the selected node in your tree.

► **To view the minimum and maximum payoffs in a tree:**

▷ Select a node, and choose **Analysis >** Payoff Range.



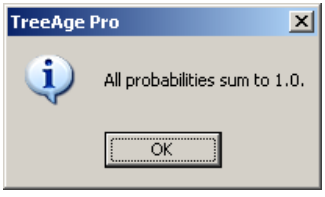
In computing the payoff range, TreeAge Pro assumes the decision maker will choose an alternative at each decision node in accordance with optimizing (i.e., maximizing or minimizing) expected value. Terminal nodes not in the optimal path are ignored.

Verify probabilities

The Verify Probabilities analysis ignores payoffs, and simply calculates the probabilities at every chance node in the tree (except those within Markov processes), reporting any problems it finds. This is a useful way to test the integrity of the probability expressions used in your tree.

► **To verify probabilities:**

▷ Choose **Analysis >** Verify Probabilities.



Errors in probabilities will be reported and problem nodes highlighted, if appropriate.

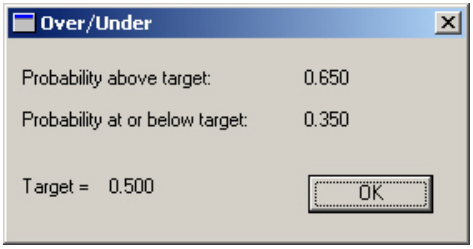
Each analysis in TreeAge Pro performs the Verify Probabilities analysis at appropriate points in the analysis. It is possible to turn off this probability error checking in a tree, as described in Chapter 22.

Over/under probabilities

The Over/Under analysis calculates the probability of achieving an outcome with a payoff over a target value, and the complementary probability of an outcome under the target.

► **To calculate the over/under probabilities:**

▷ Select a node and choose **Analysis >** Over/Under….



▷ Enter a target value. Indicate whether payoffs equal to the target value should be included in the "under" range. Press ENTER or click OK.

The resulting dialog reports the probabilities of exceeding the target value and of falling below it.

As with the Payoff Range analysis, the probability analysis includes only those outcomes which will be reached if the decision maker follows the optimal path at each decision point.

Chapter 6

Graph Windows

Each graph created in TreeAge Pro can be customized before being printed or exported. This chapter includes details on customizing the various types of graphs available in TreeAge Pro, and on using graph templates to simplify graph customization. It also explains how to export calculation results from graphs.

Users of the Excel module should refer to Chapter 28 for instructions on exporting graphs directly to Microsoft® Excel™ charts.

In this chapter:

- Customizing graphs 72
- Viewing and exporting graph data 79
- Details on specific graph types 81
- Using graph templates 85

Customizing graphs

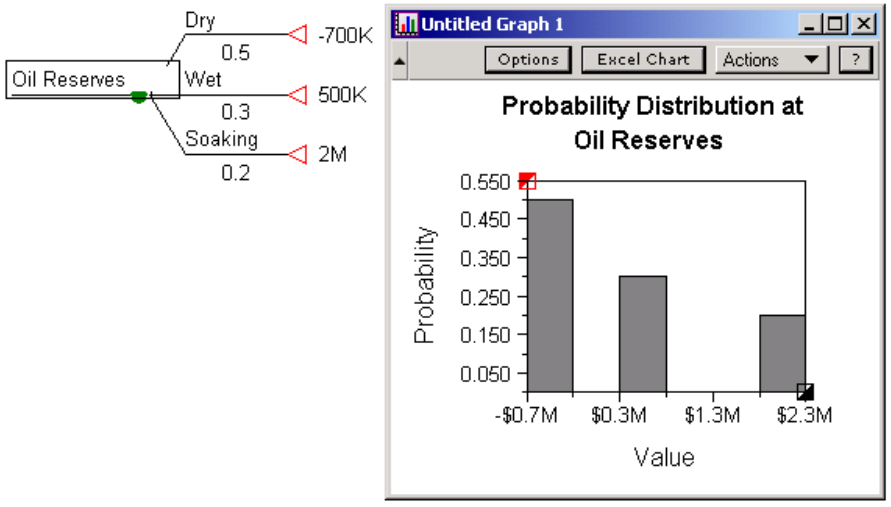
There are several kinds of changes that can be made to a TreeAge Pro graph, including modifications to labels, axes, legends, and the graph area. The methods covered in this section apply to all types of graphs created in TreeAge Pro, not just histograms. Options specific to particular graph types are covered later in this chapter, or in other chapters describing specific analyses (e.g., the section on probability distributions in Chapter 5).

To explore customization options available in TreeAge Pro, a graph is needed.

▷ Open the Oil Reserves tree, from the TreeAge Pro\Tutorial Examples subdirectory.

▷ Select the root, chance node and choose **Analysis >** Probability Distribution.

After TreeAge Pro has completed the calculations, a graph window will open displaying a histogram, like the one shown below.



Graph basics:

- When a graph window is active, a **Graph** menu is added to the menu bar.
- A tool bar inside the graph window includes an **Actions** menu with the same set of commands found in the Graph menu commands.
- Basic modifications can be made to many elements of the graph simply by clicking or double-clicking.
- Use the **Display** > Zoom commands to change the on-screen size of all elements of the graph window.

TreeAge Pro Excel module

With the TreeAge Pro Excel™ module, an option is available in most graph windows to create an Excel chart version of the graph in a blank Excel worksheet. See Chapter 28 for additional information.

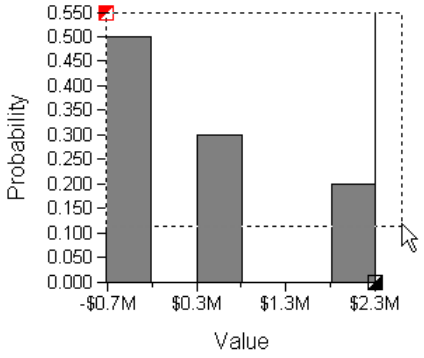
Moving and sizing the graph area

The graph area — comprised of the axes, graph lines, histogram bars, plotted points, axis labels, and the graph legend — can be moved or resized using the handles (small, half-filled squares) at the upper left and bottom right corners of the graph area.

► **To move or resize an entire graph:**

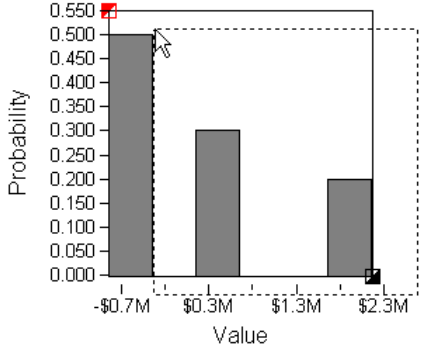
▷ To resize the graph area, click and drag the black handle at the bottom right corner.





▷ To move the graph area (relative to the window's edges and the heading), click and drag the red handle at the upper left corner.





To instead resize or move the entire graph within the printed page, use the Print Preview window for the graph, as described in Chapter 4.

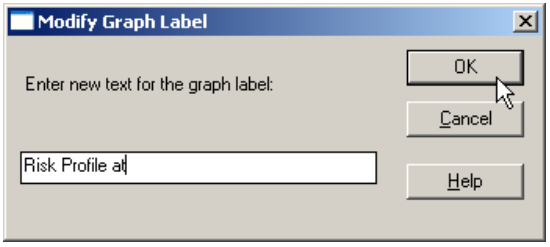
Making changes to labels

In most kinds of graphs in TreeAge Pro, three labels are created initially: a heading; an x-axis label; and a y-axis label. Each graph label can include only a single line of text; to create multi-line labels like the graph heading "Probability Distribution at Oil Reserves", text is divided among multiple labels.

It is possible to modify the text, font, and location of labels.

► **To modify a graph label's text:**

▷ Double-click on the top line of the heading, which currently reads "Probability Distribution at."



▷ In the dialog box, change the description of the analysis to `Risk Profile at`. Press ENTER or click OK.

► **To move a graph label:**

▷ Click and drag the label to a new location.



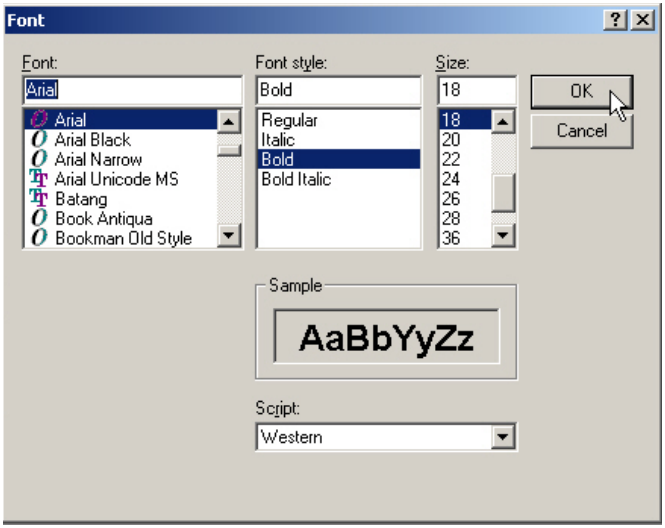
▷ After releasing the mouse button, the label is still shown as selected. To deselect the label, click anywhere in the whitespace of the graph window.

► **To modify a graph label's font:**

▷ A rectangle should appear around the text label you just edited, to indicate that it is still selected. (Click on the label once to select it, if necessary.)

▷ Select **Display** > Font, or press the Font button on the tool bar.





▷ Choose a different font, for example Arial. You might also want to experiment with changing font size and style.

▷ Click OK to apply the changes in the graph.



The selected line of the heading will reflect any changes made in the Font dialog.

Since changing the font has increased the width of the text, in this case, you can move the label box in order to re-center it over the second heading label and graph area.

Positioning and alignment of text labels:

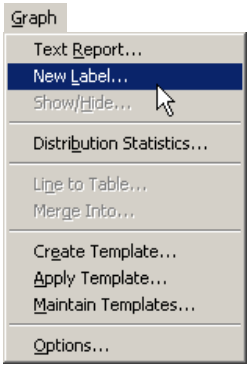
- Heading labels and custom labels (see below) are not anchored to the graph area;
- Axis and legend labels *are* anchored to the edges of the graph area (i.e., they move with the graph area);
- Although axis and legend labels move with the graph area, you can change their distance and alignment from the edge by dragging the label
- The numbers labeling axis intervals cannot be moved. However, the font, interval width, and numeric formatting of these labels can be modified in the Graph Options dialog. (See the section Making changes to numeric axes, on the next page.)

Creating custom labels

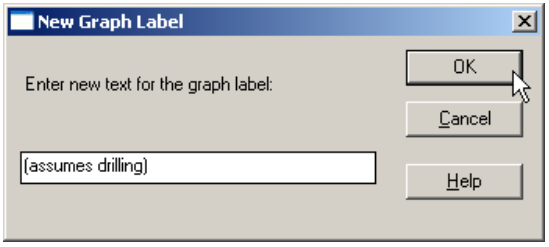
Use custom labels if you want to add notes to the graph, either inside or outside the graph area. Like other labels, these can only have one line of text.

► **To add a custom label to the graph window:**

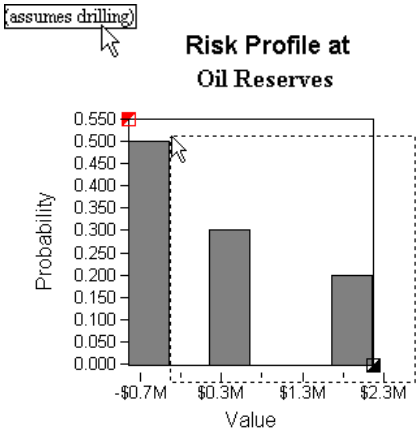
▷ In the graph window, choose **Actions >** New Label.



▷ Enter the text of your new label, and press ENTER or click OK.



▷ Change the font of the new label, if necessary, and then drag the new label to the desired location.



Making changes to numeric axes options

The numeric ranges and intervals TreeAge Pro uses in creating the graph axes can be changed either directly from the graph window or by using the Options dialog.

► **To modify a numeric axis using the Options dialog:**

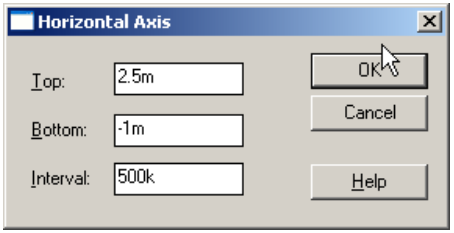
▷ In the graph window, either click the **Options** button or choose **Actions >** Options….

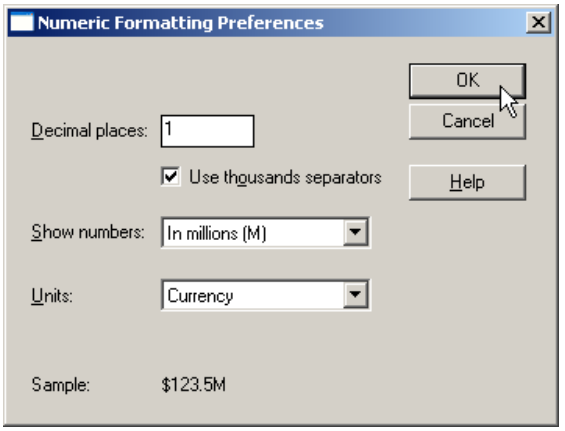
▷ In the Options dialog, select the Axes tab.

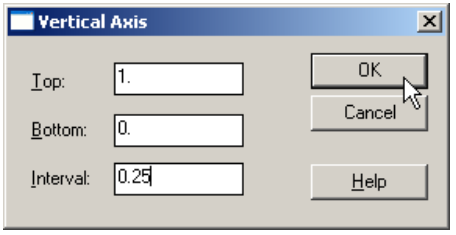


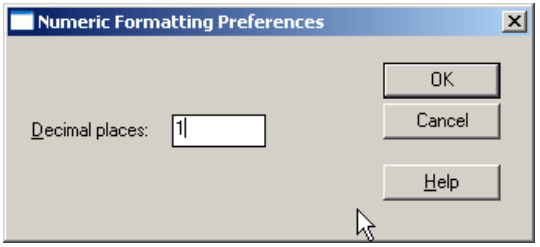
▷ Click on the Range or Format button for X-axis to change the range, intervals, and numeric formatting applied to the horizontal scale. The Y-axis buttons change the vertical axis.

▷ Each button opens an appropriate dialog box in which you can change aspects of the selected axis. The dialogs for the example probability distribution are shown here, with changed values (X-axis dialogs first, then Y-axis dialogs).









▷ If you are done making changes in the Axes options, press ENTER or click OK to return to the graph.

It is also possible to modify axes directly from the graph window, instead of using the Graph Options dialog.

► **To modify an axis scale from the graph window:**

▷ Double-click on the numbers appearing to the left of the graph's vertical axis, or below the graph's horizontal axis, to open the range dialog.

► **To change the numeric formatting of an axis from the graph window:**

▷ Select an axis by clicking once on the numbers to the left of the graph's vertical axis, or below the graph's horizontal axis. A selection rectangle should appear around the axis.

▷ With the axis selected, choose **Edit >** Numeric Formatting (or press F10).

Graph axes notes:

- In probability distribution histograms, the value axis' default number of intervals may not be ideal. Changing the range and/or interval width for an axis can be used to increase/decrease the number of intervals (as in the example graph's horizontal axis on the previous page).
- The maximum allowable number of axis intervals is 500.
- In some graphs (e.g., sensitivity analyses), an axis range may only be viewed, not changed. Charts created in Excel, using the Excel add-in, offer greater flexibility.
- See Chapter 5 for a detailed discussion of the numeric formatting options.
- In probability axes, like the vertical axis in the example, only the number of decimal places can be modified.
- While the Graph Options dialog is open, changes that you make to the range and format of the axes will be reflected in the graph window in the background.

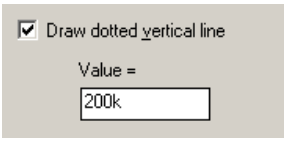
Adding a horizontal or vertical line

In some instances, you may wish to add a dotted horizontal or vertical line as a reference point. In the probability distribution graph, for example, a vertical line can be added at $200,000 to represent the expected value of the node being analyzed.

► **To display a dotted line on the graph:**

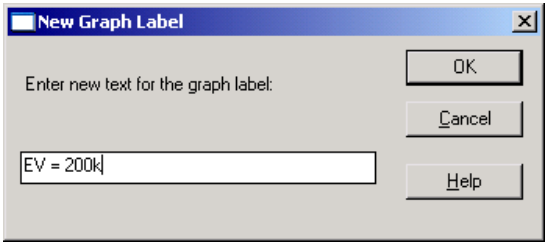
▷ In the graph window, click on the **Options** button to open the Graph Options dialog.

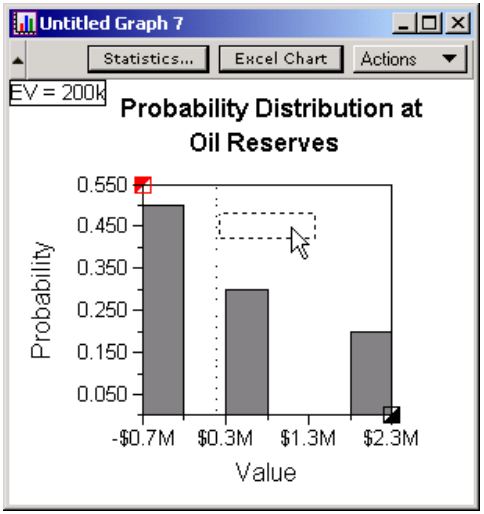
▷ In the dialog, select the Lines tab, check the option labeled Draw dotted vertical line, and type `200k` for the x-axis value.



▷ Press ENTER or click OK.

TreeAge Pro does not automatically create a label for the line. Do this manually, using the **Actions >** New Label… command, as shown above.





Viewing and exporting graph data

In addition to the numeric information displayed on the face of the graph, every graph window (with one exception) can display its underlying, calculated values. Information can be reported for the entire analysis, or for selected bars in a histogram. Particularly complex text reports may include notes to aid in understanding the results. Text reports can be easily exported to a spreadsheet or other programs.

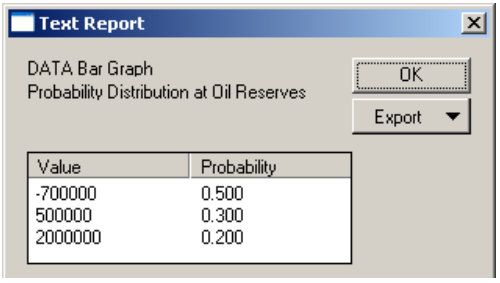
See the section on specific graph types later in this chapter for other ways of viewing a graph's underlying numbers.

Viewing a graph's text report

With the exception of animated, three-way sensitivity analysis graphs, all graphs created in TreeAge Pro have an associated text report. Also, a number of analyses (e.g., Rankings) display text reports automatically, without showing a graph first.

► **To view a graph's text report:**

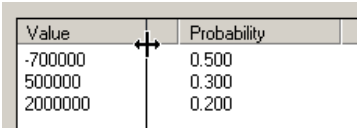
▷ In a graph window, click on the Text Report button in the tool bar, or choose **Graph >** Text Report.



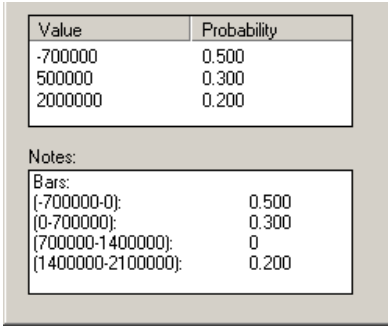
The top, main section of the text report provides the primary view of the underlying values from the analysis, and is arranged in a tabular format, with resizable columns.

► **To resize text report columns:**

▷ Place the mouse on the right edge of the column header, and click and drag to the desired width.



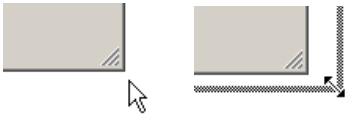
Many text reports include an additional Notes section below the main section. This section may include alternate forms or layouts of the report in the main section, and/or text explanations of the report. The Notes section can be excluded when you export the text report.



The text report dialog can be resized so that it fills more of the screen.

► **To resize the text report dialog:**

▷ Click and drag the bottom right corner of the text report dialog.



Printing and exporting text reports

While text report dialogs cannot send information directly to the printer, there are a number of ways to export information for printing or other purposes. The report is exported in tab-delimited format, making it easy to copy or import into spreadsheet, statistical, and word processing programs for printing, graphing, or further analysis.

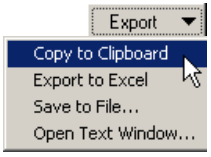
Since the width of the contents of text report columns vary considerably, the best method for printing is to export the information to a spreadsheet (or a table in a word processing document), and then adjust the column widths before printing. This can be done in a number of ways: by creating a new Excel spreadsheet directly (see below); by manually copying to the clipboard and pasting into a spreadsheet; or by saving a text file and then opening it in the other program. TreeAge Pro can also open a text-only document window showing the report.

TreeAge Pro Excel module:

With the TreeAge Pro Excel™ module, an additional option is available in the text report Export menu, allowing you to automatically export to a blank Excel worksheet. See Chapter 28 for additional information.

► **To export a graph's text report via the Text Report dialog:**

▷ Click on the **Export** pop-up menu, and choose an action.



In most graphs, the text report can copied to the clipboard without first opening the Text Report.

► **To copy a text report to the clipboard in the graph window:**

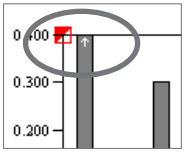
▷ Choose **Edit >** Copy Special…., click on Copy as spreadsheet-accessible text, and then press ENTER or click OK.

Details on specific graph types

TreeAge Pro creates a number of different types of graphs, including bar graphs (histograms), line graphs, and scatterplots. Many graphs can be converted into different graph types, and each graph can be customized in a number of ways. Customizations and other features that are specific to particular graph types are described below.

Bar graph

For additional details on particular types of bar graphs, see the section on probability distributions in Chapter 5, and the section on simulation strategy selection graphs in Chapter 15.



- The color or fill type used for bars can be changed by double-clicking on the fill sample in the legend and using the Modify Region Fill dialog. For more details, see the section on region graphs later in this chapter.
- If a bar's height exceeds the range of the vertical axis, a small white arrow will be displayed at the top of the bar. This arrow may be hard to see, however, if a bar does not use a solid color. (See above for details on how to change the axis range or other characteristics.)
- The most common use of bar graphs in TreeAge Pro is for displaying probability distributions (e.g., of outcomes or samples values), in which the horizontal axis is numeric. Changing the *range* of the horizontal axis may cause some bars to not be displayed; changing the *interval width* may cause both the number and height of bars to change.
- TreeAge Pro auto-scales the vertical, probability axis in histograms when a new graph is created or the height of the highest bar changes (usually due to changing the interval width on the horizontal axis).
- In a few analyses, a bar graph is created with a non-numeric horizontal axis (for instance, in a Monte Carlo simulation strategy selection graph, each strategy gets a vertical bar).

Line graphs (and cost-effectiveness graphs)

In line graphs (including comparative distributions and one-way sensitivity analyses), each option is represented by a line connecting the series of points defined at intervals. For easy identification, each option uses a different style of line marker. It is possible to change the marker used for an option (including using no marker), and/or change the color of the line itself. You can also selectively hide lines in the graph.

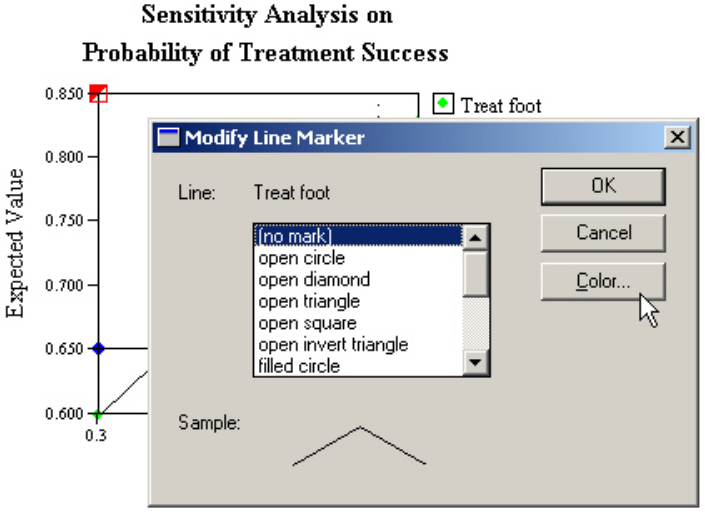
► **To turn the display of selected lines off or on:**

▷ Choose **Actions** > Show/Hide.

▷ In the dialog's list of alternatives, click on a highlighted line to de-select it (i.e., hide it). To show a currently hidden line, re-select it in the list.

► **To change the marker used for an option:**

▷ Double-click on the appropriate marker symbol **in the legend** to the right of the graph area.



▷ In the Modify Line Marker dialog, select a marker from the list displayed in the resulting dialog box. Click OK.

If you choose "(No Mark)" from the list, you may also want to change the line to use a color other than black.

► **To change the color of a line:**

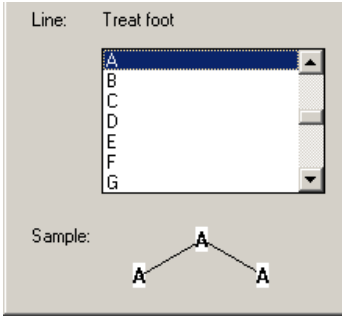
▷ In the Modify Line Marker dialog, click on the Color… button.

▷ In the Color dialog, either select a pre-defined color or define a custom color. Click OK to return to the Modify Line Marker dialog.



Notes on line markers and colors:

- In cost-effectiveness graphs, only line markers can be modified; see Chapter 31 for more information on cost-effectiveness graph options.
- In graphs with more than a few strategies, it may be useful to change the line markers for all options to use alphabetical markers, perhaps starting with "A" for the top strategy.



Any line in the graph may be converted into a table for subsequent use in formulas (instead manually copying the report to a spreadsheet, creating a table in TreeAge, and then copying a column into the table).

► **To create a table from a graph line:**

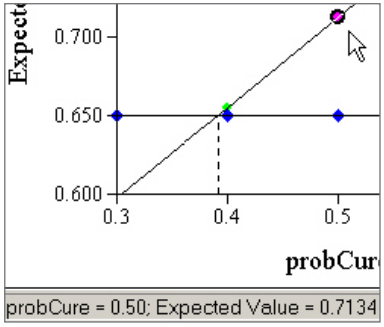
▷ Choose **Actions >** Line To Table, and enter the desired table name and other properties.

See Chapter 18 for more information on creating and using tables in TreeAge Pro.

In addition to using the text report to view all of the results underlying a line graph, it is also possible to look at individual points in the graph.

► **To display the coordinates of a point on a line:**

▷ Hold down the CONTROL key and place the cursor close to a line.



A gray dot will appear on the closest point along that line, and the approximate x- and y-axis coordinates of the gray dot will appear in the status bar.

Scatterplots

The color of an option's points in a scatterplot can be modified exactly as described above for line graphs.

The Graph Options dialog also includes settings to change the size of the points used in the plot.

See Chapter 33 for information on particular scatterplot formats.

Tornado diagrams

In a tornado diagram, the process for changing the pattern used to fill each bar is identical to that described for region graphs.

And similar to a line graph, it is possible to hide selected bars in a tornado diagram using the Show/Hide command. See Chapter 15 for more details on tornado diagrams.

Region graphs

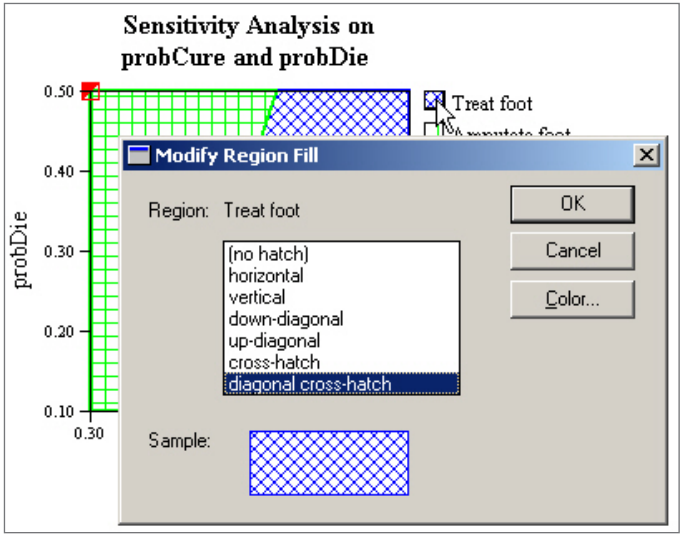
Region graphs are generated from two- and three-way sensitivity analyses. A strategy graph, generated from a one-way sensitivity analysis, is also a type of region graph.

► **To change the pattern used to fill each region:**

▷ Double-click on the marker as it is displayed in the legend to the right of the graph. Clicking on a region inside the graph will not work.

▷ Select the desired hatch pattern from the list displayed in the dialog box, or select "no hatch" for a solid region.

▷ You may optionally change the color used for drawing regions by pressing the Color button.



▷ Press ENTER or click OK.

Using graph templates

Any number of graph templates can be created to store display preferences and other information about the appearance of a TreeAge Pro graph. One or more templates can then be applied to new graphs, making it much easier to maintain consistency of fonts, numeric formatting, graph area size, and text labels.

What a graph template can contain

Each graph template automatically contains the size of the graph area. Additional options that may be available include (some may be disabled if they do not apply to the active graph window):

- **Title labels** – Graphs usually have two title labels; one or both can be stored. For example, you may prefer all probability distribution titles to read "Risk Profile at" on the first line. If only one title line is included in the template, the other label will be created dynamically, as usual.
- **Axis labels** – The template may include either or both axes' text labels.
- **Custom labels** – The text, positions, and fonts or all labels added with the New Label command can be included in the template.
- **Fonts** – The font settings to apply to text in the graph.
- **Numeric formats** – The numeric formatting to use with graph axes.
- **Custom lines** – Available if the graph options specify a custom dotted line be displayed in your graph.

Creating and applying graph templates

Graph templates are created based on an active graph window. After customizing a graph to have the appropriate appearance, then you can create a template based on the graph.

► **To create a graph template:**

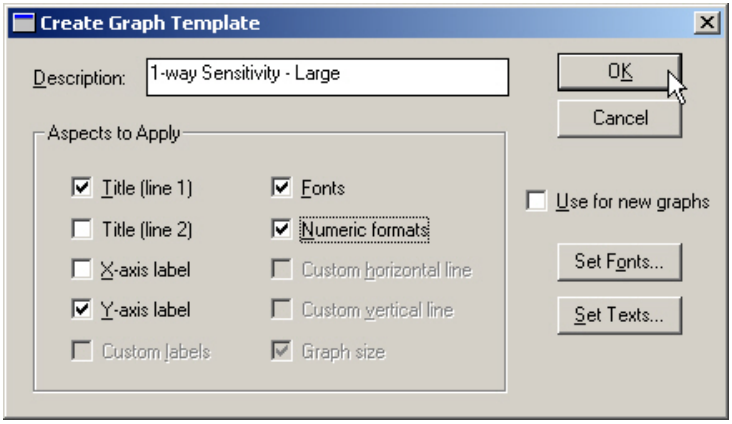
▷ Open or create and customize a graph, then choose **Actions >** Create Template.

▷ Give the template a description. This is for your reference, when choosing a template to apply to a new graph.

▷ Check the appropriate boxes in the list to indicate those items to apply when this template is used on a new graph.

▷ To change the fonts or title and axis texts that will be included in the template, click the Set Fonts… or Set Texts… buttons.

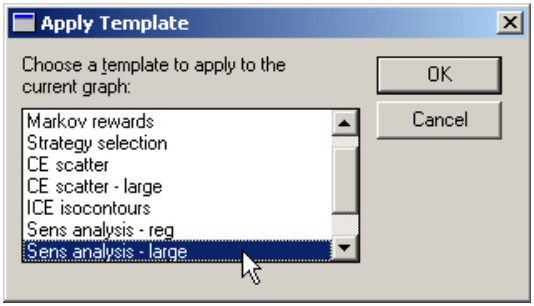
▷ Click OK or press ENTER to apply the changes.



► **To apply a template to a graph:**

▷ Create a new graph or open an existing graph, and choose **Actions >** Apply Template.

▷ Select the desired template from the list. Click OK or press ENTER to update the graph.



When a template is applied to a graph, only the selected settings will be used. However, even unselected settings and items are stored with the template. It is possible to edit an existing template later (see below), and change which options will be applied.

Creating a default graph template

A particular graph size and font settings can be automatically applied to every new graph you create. Unlike normal graph templates, the default template will apply only font and graph size information.

► **To set a default graph size and font for new graphs:**

▷ In a new or existing template, select the option labeled Use for new graphs.

If you need to change the default graph size and font settings, create a new template with the desired attributes and select it as the template to use for new graphs.

Modifying an existing graph template

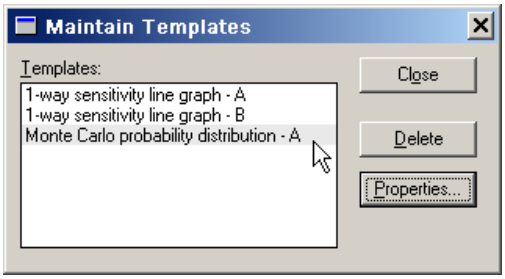
It is possible to change which of an existing template's attributes are applied to graphs.

► **To edit a graph template's properties:**

▷ Open a graph window (in order to activate the Graph menu).

▷ Choose **Actions >** Maintain Templates.

▷ Select the template from the list, and press the **Properties…** button.



You may also indicate that the template's graph size and fonts should be the default used for new graphs.

Using graph templates with stored analyses

A graph template can be applied automatically to graphs created from a stored analysis. See Chapter 19 for a full description of running stored analyses.

Chapter 7

Tree Calculation Methods and Preferences

Any single tree may be calculated and evaluated in a variety of ways, simply by switching between different payoffs, or from single- to multi-attribute calculations. This chapter covers tree calculation method preferences, including two kinds of multi-attribute calculation methods.

An additional multi-attribute calculation method, Cost-Effectiveness, is available to users of the Healthcare module; see Chapter 31 for details.

Other chapters cover the other preferences categories; see Appendix C for an overview of all preferences.

In this chapter:

- Changing what the tree calculates 88
- Calculation using multiple attributes 93
- Roll back analysis options 95
- Regional/international numeric settings 97
- Overriding the optimal path at selected nodes 98

Changing what the tree calculates

Some basic tree calculation preferences were introduced at the end of Chapter 2 and the beginning of Chapter 5:

- the calculation method
- the active payoff(s)
- the optimal path criterion
- numeric formatting

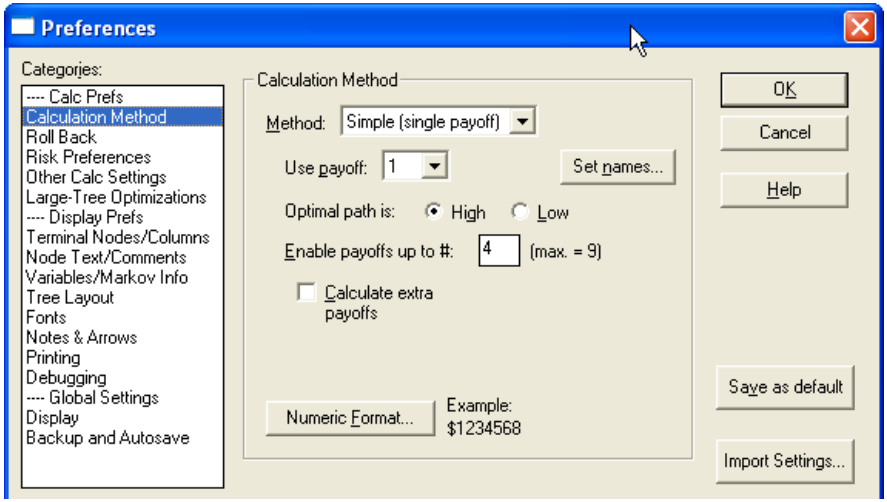
Some information about the currently selected calculation method can be seen in the status bar at the bottom of the TreeAge Pro window. If the tree is currently set to use only one payoff — the Simple calculation method — "Payoff *n*" will appear, where *n* is the active payoff set.



To view in detail the tree's current calculation method, or to make changes to these settings, open the Preferences dialog.

► **To view/edit the calculation method preferences:**

▷ Choose **Edit >** Preferences…. By default, the Calculation Method preferences category will be visible.



Tip:

The **Import Settings** button allows all preferences in the current tree to be replaced by the preferences from a different tree file. If the "Save all settings…" checkbox is selected, the current tree's preferences will be saved as the defaults for all new trees when you click **OK**. See Appendix C for an overview of the Preferences dialog.

Selecting calculation method

The choice of calculation method determines the formula used to calculate values for nodes in your tree. There are two kinds of calculation methods:

- Simple, single-attribute calculations;
- Multiple-attribute calculations, including: Benefit-Cost and Multi-Attribute (weighted).
- Healthcare module users can use a third form of multi-attribute calculations, Cost-Effectiveness.

Simple calculations are just that — expected values are calculated for the nodes in the tree based simply on the values in the active payoff set. If, as described above, your tree included *multiple* attributes — such as monetary benefits in payoff #1 and costs in payoff #2 — the two sets of payoff values could be combined in a single calculation using the Benefit-Cost calculation method's formula.

► **To change the calculation method:**

▷ Roll back must be turned off.

▷ With the Preferences dialog open, select the appropriate calculation method from the Method pop-up menu.

▷ If you select a calculation method for the first time in a particular tree, the settings for payoffs, optimal path criterion, and numeric format may need to be modified.

More information on the multi-attribute calculation methods is provided later in this chapter.

Calculation method notes:

- Changing calculation method (or other) preferences will not affect the content of your tree. Values or formulas already entered in payoff #1 **will not be lost** by switching the active payoff set from #1 to #2, for example. Only the calculation and display of values will be affected; the changes to the calculation preferences can be reversed at any time.
- When the Preferences dialog is closed, TreeAge Pro updates the information in the the status bar. If you have switched Simple calculations to use payoff 3, "Payoff 3" will appear in the status bar. If you select Benefit-Cost using payoffs 4 and 2, the status bar will read "B-C, 4-2." If you select Multi-Attribute, then "MultiAttr" will show in the status bar.

Use payoff *N*

In TreeAge Pro, you can enter up to nine different payoffs at each terminal node. Sets 5 through 9 are disabled in the Enter Payoff window, by default, but can be activated in the Calculation Method preferences (see below). In a Simple analysis, you can use any one of the payoff/reward sets.

► **To change the active payoff:**

▷ In the Calculation Method preferences, with the Simple method selected, select a payoff set in the Use payoff pop-up menu.

▷ When you change to a new active payoff for the first time in a tree, ensure that you select the appropriate optimal path criterion and numeric formatting for that payoff.

If you select a multi-attribute calculation method, you can select which payoff sets to use for each attribute.

In a benefit-cost model, for example, monetary benefits could be assigned to payoff #1 and monetary costs to payoff #2 (or vice versa). You would then have the option of performing simple, single-payoff analysis on payoff #1 or payoff #2 or, as will be explained in a later section, of performing multi-attribute analysis.

Optimal path

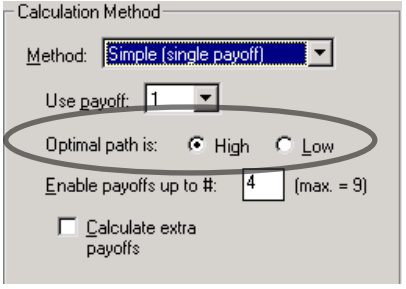
All calculation methods in TreeAge Pro require you to specify how decisions should be made in the tree (i.e., during roll back). This is done using the option labeled Optimal path is.

The default optimal path criterion used for new trees is High, unless you save new default settings. This setting would be changed if, for example, you were modeling project costs (to find the strategy with the lowest expected cost).

► **To set the optimal path criterion:**

▷ Choose **Edit >** Preferences. The Calculation Method category will be visible.

▷ Next to the label Optimal path is, select the appropriate setting: High for a tree that should maximize expected values, Low for a tree that should minimize.



For a tree set to High (e.g., one whose payoffs are in terms of life expectancy), at each decision node TreeAge Pro will select the alternative with the highest numeric value. For a tree that minimizes, such as one whose payoff formula is in terms of costs, the alternative with the *lowest* numeric value is selected.

> ***Note:***
> Each of the nine payoff sets stores its own optimal path criterion for Simple calculations, and each form of multi-attribute calculations also has its own optimal path setting.

Enable payoffs up to

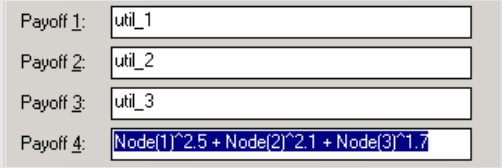
In TreeAge Pro, you can enter up to nine different payoffs sets (attributes) in a tree. Sets 5 through 9 are disabled in the Enter Payoff window, by default, but can be activated by increasing the number in this preference.

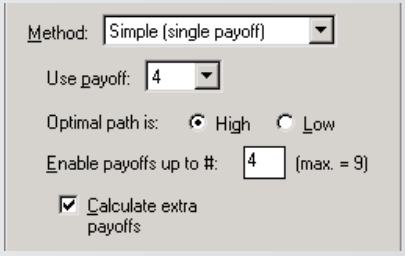
Conversely, it is possible to deactivate (gray out) additional payoff sets, for example only leaving set #1 active.

Another use for this setting is in conjuction with the next setting, Calculate extra payoffs.

Note:

If appropriate for your model, you can use the Calculate extra payoffs setting and Node() function to create a custom calculation method. If the active payoff set will reference other payoff sets using the Node() function, make sure the active payoff set(s) are beneath (i.e., use higher set number than) the extra payoffs, as shown here.

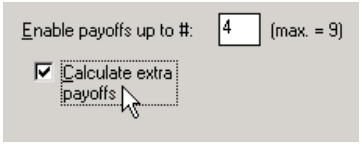




Calculate extra payoffs

It is possible to calculate the tree and optimize at decision nodes based on the current active payoff method and set(s), but report expected values for other payoff sets. For example, you could roll back the tree based on minimizing costs in payoff #1, but also report expected values for some types of utility attributes which you have entered in payoffs #2 and up.

Check the box labeled Calculate extra payoffs, and adjust the Enable payoffs… setting to control how many extra payoff sets will be calculated.



To report extra payoffs in roll back, set up terminal node columns using a custom calculation like "Node(*N*)" where *N* is the extra payoff. The Node() function can be used in a variety of ways to retrieve expected values; see Appendix D for information on the Node() function.

Monte Carlo simulations automatically include the calculated extra payoffs using additional text report columns and distribution graphs. The detailed Markov analysis text report will include the extra reward sets.

- If you activate the calculation of extra payoffs, you must fill in the enabled payoff sets at all terminal nodes, even though they are not specifically required by the current calculation method.
- Under Multi-attribute calculations, weight values must be specified for every enabled extra payoff. Use a 0 weight to exclude an extra attribute from the regular, weighted expected value calculation (this will **not** 0 the extra payoff's reported value).

Numeric formatting

The value displayed next to the **Numeric Format…** button at the bottom of the dialog shows how calculated values will be formatted during roll back and a variety of other analyses, based on the current settings. The default settings are two decimal places, with no abbreviations, using your computer's currency settings.

The settings can be modified via the Preferences dialog, by clicking the **Numeric Format…** button (or under **Edit >** Numeric Formatting). See Chapter 5 for details on the Numeric Formatting dialog.

Calculations using multiple attributes

While many analyses will require only one attribute (e.g., cost), some models may have multiple attributes (e.g., benefits and costs) or different perspectives on the same attribute (e.g., societal, personal, or institutional costs of disease).

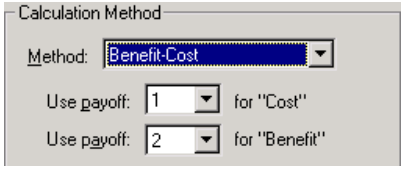
TreeAge Pro includes two basic calculation methods that combine multiple payoffs: benefit-cost and weighted multi-attribute calculations. (If you have the optional Healthcare module, a third calculation method using multiple payoffs is available: Cost-Effectiveness. See the Healthcare module documentation for details.)

The first step in preparing a multi-attribute model is setting the calculation preferences.

► **To select a multi-attribute calculation method:**

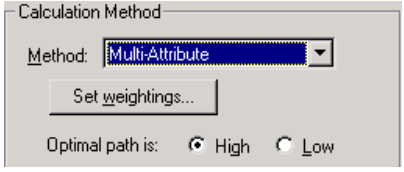
▷ Choose **Edit >** Preferences.

▷ Change the Calculation Method pop-up menu selection from Simple to one of the multi-attribute options.



▷ If you select Benefit-Cost, select two payoff sets to represent the two attributes in your tree. The optimal path will automatically be High (maximize value).

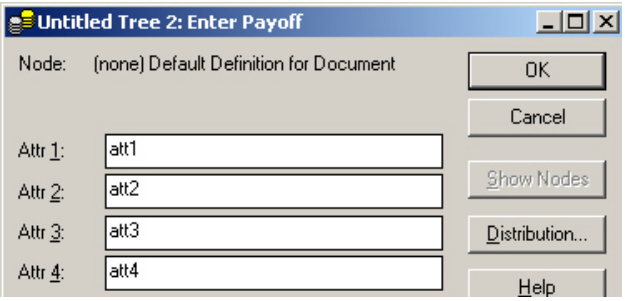
▷ If you select Multi-Attribute, click the **Weightings…** button to enter weightings (details are provided below), and select the optimal path criterion.



▷ Click **Numeric Format…** to change the formatting applied to values calculated using the relevant multi-attribute formula.

Entering multiple payoffs

With both kinds of multiple-attribute models — benefit-cost and multi-attribute — you will enter at least two payoffs at each terminal node. The process of entering multiple payoffs values at a terminal node is very simple — the payoff sets in the Enter Payoff window correspond to the Use payoff selections made in the Preferences dialog.



If a multi-attribute calculation method is used, the payoff titles in the window will indicate which payoffs are to be used. For benefit-cost, they will be labeled "Benefit" and "Cost." For weighted multi-attribute models, they will be labeled "Attr 1" through "Attr 9."

If you require more than four separate payoff sets in your model, for instance for weighted multi-attribute calculations, additional payoffs up to #9 can be activated by changing the Enable payoffs… setting.

How benefit-cost calculations are performed

This calculation will subtract the cost of a scenario, represented by one payoff, from its benefit, represented by a different payoff. Therefore, both attributes in a benefit-cost analysis must be measured in the same monetary units. All analyses available under Simple calculations (1-, 2-, 3-way sensitivity analysis, for example) are available under the Benefit-Cost calculation method, as well.

This calculation does not divide costs by benefits, as is done in some forms of cost-benefit analyses. To create a custom cost/benefit analysis calculation method, refer to the section on the Calculate extra payoffs preference and the Node( ) function.

How weighted multi-attribute calculations are performed

Unlike the Benefit-Cost calculation method, the weighted Multi-Attribute calculation method does not require that all attributes/payoffs be assigned using the same units. Instead, a linear equation is set up that combines up to nine payoffs; attributes given a non-zero weighting are made part of the multi-attribute calculations.

For example, if you assigned a weight of 1 to Attribute 1 and a weight of 0.5 to the next three attributes, each terminal node would be evaluated based on the expression *Attribute 1 + 0.5 * Attribute 2 + 0.5 * Attribute 3 + 0.5 * Attribute 4*.

Variables (see Chapter 12) can be used in the weighting expressions. This is useful when there is uncertainty concerning how much one factor should be weighted versus another.

The Weightings dialog can be accessed either by clicking the Set Weightings… button in the Calculation Method category of the Preferences dialog, or by choosing Values > Multi-Attribute Weights from the tree window. All analyses available under Simple calculations are available under Multi-Attribute calculations, as well.

Roll back analysis options

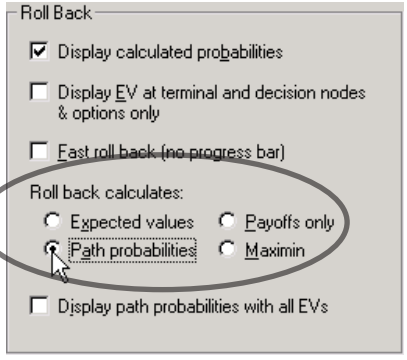
Before dealing with multiple-attribute analysis methods, let's look at some calculation preferences related to roll back.

Typically, roll back boxes display the expected value of each node, along with the path probability at terminal nodes. TreeAge Pro can display a variety of other calculated values during roll back; these are described below.

► **To change the quantity to be calculated during rollback:**

▷ Choose **Edit >** Preferences, and select the Roll Back category.

▷ Select an option from the group labeled Roll back calculates.



Here is a brief description of the other options:

Payoffs only: Only the values of terminal nodes will be displayed. Optimal paths will be indicated using hash marks and colored lines, but no expected values will be displayed.

Path probabilities: The display of expected values is suppressed. Path probabilities are calculated for all terminal nodes (does *not* take into account the optimal path). If there are any decision nodes, the sum of the path probabilities of all terminal nodes will be greater than 1.0.

Maximin: This option will consider only the most pessimistic possibility at each uncertainty, regardless of probabilities. Then, at each decision point, the best option is selected. More details are provided in the following section.

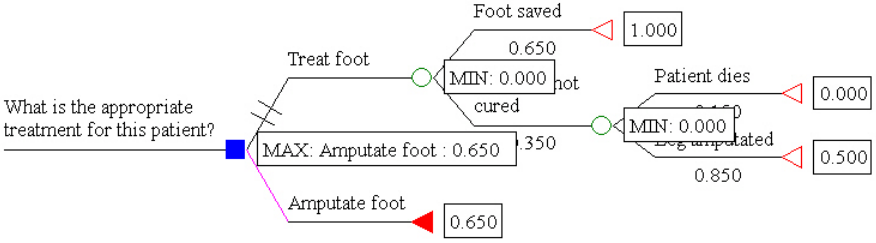
Maximin (and Minimax) roll back

Maximin calculations may be useful during early stages of certain models, before probabilities have been assigned to the tree. Here, specifically, is how a tree is rolled back under Maximin:

- The value assigned to every chance node is equal to the worst (least optimal) value of any of its potential outcomes. *Probabilities are ignored*, and may be left blank.
- The value assigned to every decision node is equal to that of the best alternative, as usual. (*Note: If you use the Options > Reverse Optimal Path command at selected decision nodes, these nodes will work like chance nodes, as described above.)*

Maximin calculations take a pessimistic view of events. They are based on the idea that one way to deal with risk is to identify the worst case scenario for each alternative and then select the strategy which yields the best of these worst case scenarios.

Below is a picture of the example file Rock Climber using Maximin roll back. The value of the Treat foot strategy is simply its worst outcome, which is the patient dying (utility = 0). The Amputate foot option has no uncertainty, and it is simply equal to its payoff. The decision then maximizes between the available options, as usual.



In a Maximin roll back of a tree whose optimal path preference is set to High, the roll back boxes at chance nodes will be labeled MIN: at decision node the boxes will be labeled MAX:, in order to indicate the operation being performed. These prefixes are switched if the optimal path preference is set to low (minimization).

Regional/international numeric settings (decimals and thousands separators)

By default, the regional numeric settings specified in your operating system determine which character TreeAge Pro will recognize as a decimal separator when you enter numbers — usually either a "." or a "," — and which character, if any, represents a thousands separator.

For example, your computer's settings are used to decide whether the entered number:

```
1,375
```

should be interpreted as

```
1 + 375/1000     (comma = decimal)
```

or as

```
1375         (comma = thousands separator)
```

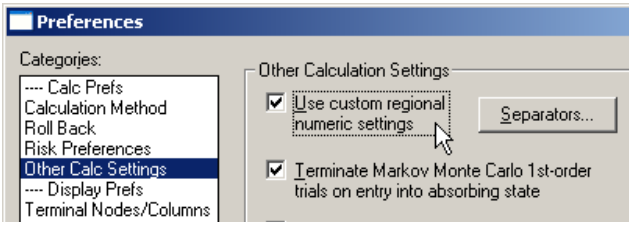
The operating system's regional settings are also, by default, used by TreeAge Pro to determine what decimal and thousand separator characters are used in displaying calculated values.

However, TreeAge Pro includes preferences that enable particular trees to override the operating system's regional numeric settings, allowing you to specify which characters to use for decimal and thousands separators.

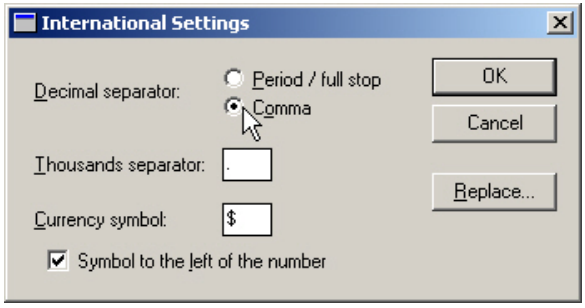
► **To enable custom regional numeric settings in a tree:**

▷ Choose **Edit >** Preferences…, and select the Other Calc Settings category.

▷ Select the option Use custom regional numeric settings, and click the **Separators…** button.



▷ In the International Settings dialog, modify the Decimal separator, Thousands separator, and Currency symbol settings as necessary.



▷ Press ENTER or click OK to close the International Settings dialog, and again, to close the Preferences dialog.

These settings will apply when this particular model is opened, modified, and analyzed *on any computer, no matter what the operating system's regional numeric settings*.

Overriding the optimal path at selected nodes

TreeAge Pro selects an optimal path at decision nodes based on the calculation method preferences covered in previous chapters. At chance nodes, multiple branches are included in the expected value calculation. There are a number of ways to override these behaviors at selected nodes, however.

Force path

The Force Path command lets you force a particular outcome at a chance node, or a particular choice at a decision node, by converting to a logic node. Choosing **Options >** Force Path at a branch:

- converts the selected node's parent to a logic node; and
- sets the logic condition/probability of the selected branch to 1 and the other branches to 0.

When you have a complex tree and wish to indicate that a previously uncertain event has taken place, select the branch that represents that path and choose Options > Force Path.

It is recommended that you save a copy of your tree before forcing the path at a node, so that you can easily recover the tree's original state.

Change optimal path

At a selected decision nodes, you can force TreeAge Pro to apply the opposite optimal path criterion to that specified in the tree's preferences.

► **To change the optimal path for a given decision node:**

▷ Select the node.

▷ Select **Options >** Change Optimal Path.

▷ Click Yes in the confirming dialog.

The decision node will reappear with an arrow inside it. The arrow will point upward if that decision node has been reset for maximization, or downward if the node has been reset for minimization. If you reverse the optimal path for an entire tree in which the optimal path of one or more nodes has been individually changed, the arrows in all of the individually changed nodes will reverse, signifying that they remain different from the rest of the tree.

The reversal of a node's optimal path can be undone by selecting the node and then performing the operation described above a second time.

Chapter 8

Selecting Subtrees and Multiple nodes

While some tree-building tasks require that a single node be selected, other operations can, or must, be performed on multiple nodes. This chapter describes the different methods used to select multiple nodes.

In this chapter:

- Selecting a subtree 100
- Selecting multiple, unrelated nodes 100
- Selecting multiple nodes by characteristic 101

Selecting a subtree

In decision trees, a *subtree* refers to a part of a tree, defined in this way: starting with a node with branches, that node's subtree is comprised of all nodes and branches to its right (its "descendants").

A few tree-building operations — copying, pasting, cutting, clearing, and cloning — require a special method for selecting a *subtree*.

► **To select a subtree:**

▷ Select a single node that has branches attached (i.e., not a terminal node).

▷ Choose **Options >** Select Subtree.

It is also possible to select a subtree without using the menus.

► **To select a subtree:**

▷ While holding down the CONTROL key, click on the subtree's root node.

Note: The methods of selecting multiple, individual nodes described in the next sections cannot be used to select a subtree.

Selecting multiple, unrelated nodes

TreeAge Pro offers several methods for selecting multiple, unrelated nodes. These techniques can be used when changing node types, entering payoffs, and a few other tasks.

► **To select a set of unrelated nodes:**

▷ Select any node.

▷ While holding down the SHIFT key, select another node by clicking on it. Continue adding to the selection using the same shift-clicking operation.

► **To remove nodes from the selection:**

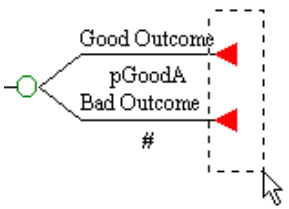
▷ Hold down the SHIFT key and click, one after the other, on each of the selected nodes that you wish to deselect.

It is possible to select multiple nodes by dragging a selection rectangle around them.

► **To select several adjacent nodes:**

▷ Click and drag to create a selection rectangle that encloses the adjacent nodes you wish to select.

▷ Release the mouse button.



You may also use the selection rectangle described above to add or remove nodes from the current selection, by holding the SHIFT key.

Selecting multiple nodes by characteristic

The Select Nodes dialog makes it possible to automatically select sets of nodes that share a particular characteristic, such as:

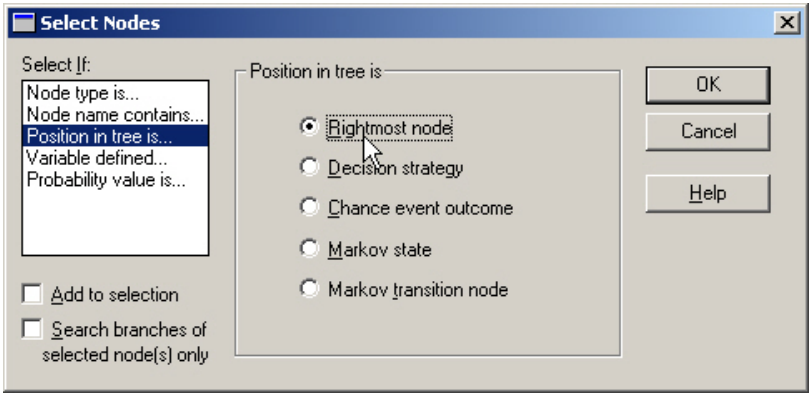
- a character, word, or phrase in the node text
- the location of the node
- a value used or defined at the node
- a node's type, or its parent's type

A common application of this feature is to select all rightmost nodes in the tree, prior to performing the Change Node Type command — creating all endpoints at once. This selection is illustrated below.

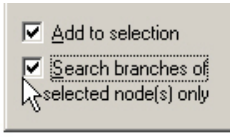
► **To select a set of nodes based on a shared characteristic:**

▷ Choose **Options >** Select If….

▷ Click on a category in the Select If list, and specify the selection criteria.



The Select Nodes dialog includes an option to add to the current set of selected nodes; otherwise, any initially selected nodes will be unselected. The search can also be restricted to particular subtrees.



(Note: Text searches are not case-sensitive. For more advanced text search options, use the **Options >** Find/Replace dialog.)

Chapter 9

Making Changes to Tree Structure

Building a tree is often a process of frequent revision. Also, new tree projects can often make use of subtrees from existing models. This chapter starts with a review of the basic tree building commands from Chapter 2, and then covers a set of features designed to help you move, remove, and duplicate parts of a tree (called "subtrees").

In this chapter:

- Tree building commands – a review 104
- Inserting, deleting, and reordering branches 105
- Cut, copy, paste, and clear nodes and subtrees 108
- Cut, copy, paste, and clear text 111
- Undo and Redo 112
- Reverting to a file's saved state 112
- Find and replace text, formulas, and values 113
- Using the Probability Wheel 114

Tree-building commands – a review

Here is a quick review of some of the basic tree-building commands introduced in Chapter 2.

► **To add branches:**

▷ Select a node.

▷ Choose **Options >** Add Branches (or double-click on the node).

► **To delete a branch:**

▷ Select the branch to remove.

▷ Choose **Options >** Delete Branch.

► **To change a node's type:**

▷ Select a node.

▷ Choose **Options >** Change Node Type… (or click on the Change Node Type button).

▷ In the Change Node Type dialog, select the desired node type, and press ENTER or click OK.

Note that these tree-building commands are all found under the Options menu. The next section describes in more detail how the Delete Branch command works, and also introduces two other useful commands found under the Options menu: Insert Branch and Reorder Branches.

This chapter also details the commands used to add and delete entire sections of a tree, rather than single nodes.

A note on tree terminology

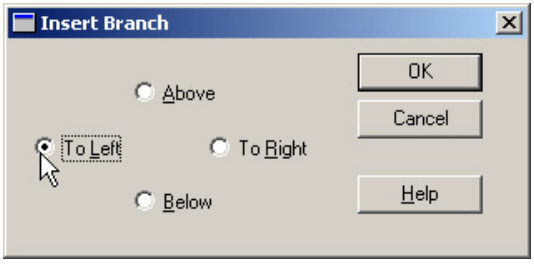
In tree graphs, familial terms are often used to help identify particular nodes and branches based on their relative locations. Assuming a reference node:

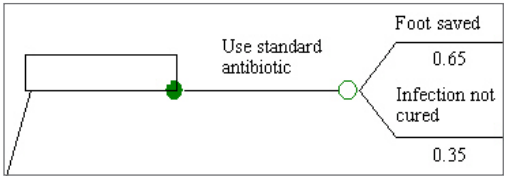
- nodes in its path back to the root node are its **ancestors**;
- nodes in the paths to its right are its **descendants**;
- branches of a **parent** node are its **children**;
- branches of a parent node are also each other's **siblings**.

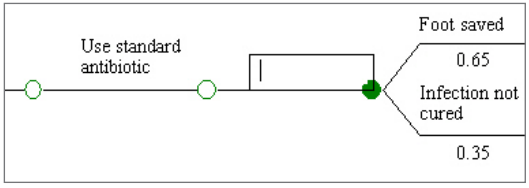
Inserting, deleting, and reordering branches

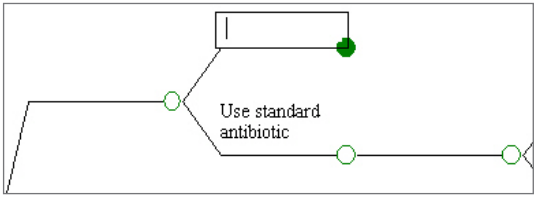
Many revisions to a tree's structure can be accomplished by inserting or deleting one branch or node at a time, or by changing the order of branches emanating from a node. These commands are described in detail in this section.

The Cut, Copy, and Paste commands described later in this chapter can be used to efficiently carry out similar, but more complex, tree modifications – including inserting, deleting, and reordering *subtrees*, as well as duplicating nodes or subtrees.









Inserting Branches

The Insert Branch command, in comparison to the Add Branches command, provides greater control over where new branches and nodes are created. While the Add Branches command can only add more children to a node, the Insert Branches command can be used to either add branches or make "generational" changes — i.e., to insert a new node between an existing parent and its children. Inserting a node causes existing nodes to *move* – up, down, to the left, or to the right.

► **To insert a branch:**

▷ Open the Rock Climber tree, from the TreeAge Pro Tutorial Examples folder.

▷ Select the chance node labeled Use standard antibiotic.

▷ Choose **Options >** Insert Branch….

▷ Click To Left, and press ENTER or click OK.

A new branch, ending in a chance node, is inserted between the root, decision node and the Use standard antibiotic node.

▷ Select the Use standard antibiotic node and choose Options > Insert Branch….

▷ Click To Right, and OK.

This time the new branch is inserted after the Use standard antibiotic node. The two branches that were the children of the Use standard antibiotic node are now attached to the new chance node.

▷ Again, select the Use standard antibiotic node and choose Options > Insert Branch….

▷ Click Above, and OK.

In this case, the new branch is added as a sibling of the selected, Use standard antibiotic node.

Choosing the Below radio option, instead, would have added the new sibling below Use standard antibiotic.

Using the Insert Branch command at a node with no branches works just like the Add Branches command, except that it adds one branch instead of two.

> ***Tip:***
> The Delete Branch command is used in this situation for illustrative purposes. The Undo command, covered later in this chapter, may be more efficient when you need to reverse the sequence of changes you have just made. The Delete Branch command is more flexible, however, as it allows you to reverse changes that were made at any time in the past, and leave in place more recent revisions.

Deleting branches

To illustrate the different ways the Delete Branch command can be used, let's now reverse the changes made using the Insert Branch command.

The Delete Branch command can be used to remove any node, except the root node.

► **To delete an end node:**

▷ Select the blank node above the Use standard antibiotic node.

▷ Choose **Options >** Delete Branch.

The selected node has been deleted. The tree again looks like the picture at the top of the previous page.

The next step is to remove the blank node inserted between the Use standard antibiotic node and its original set of branches.

► **To delete an internal node:**

▷ Select the blank node to the right of the Use standard antibiotic node.

▷ Choose **Options >** Delete Branch.

The deleted node's children move one generation to the left, joining any siblings of the deleted branch. To see another example of this, delete the blank node to the left of the Use standard antibiotic node.

▷ Select the blank node to the left of the Use standard antibiotic node.

▷ Choose **Options >** Delete Branch.

The Rock Climber tree should now be back in its original state. If not, you can either use the Undo command repeatedly, or close the file without saving changes and then re-open it.

Reordering branches and inserting sets of branches

The Reorder\Manage Branches dialog can be used to quickly change the order of a node's branches.

► **To reorder a node's branches:**

▷ In your tree, select a single node.

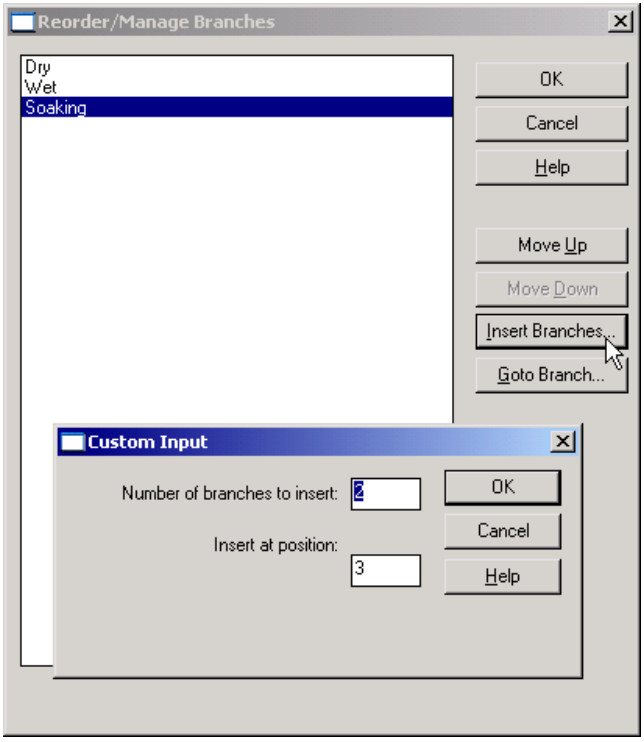
▷ Choose **Options >** Reorder\Manage Branches….

The resulting dialog box lists the branches of the selected node in order, from top to bottom.

▷ Select one of the branches in the list, and click Move Down or Move Up.

▷ Press ENTER or click OK.

The dialog can also be used to jump to a selected branch in the list, or to insert a set of branches anywhere in the branch order.



Cut, copy, paste, and clear endnodes and subtrees

Frequently, sections of a tree can be reused in another part of the same tree, or a different tree that you are working on. TreeAge Pro allows you to select an endnode or an entire subtree, copy or cut it, and then paste it to one or more nodes in any open tree.

A different node or subtree can be copied or cut onto each of TreeAge Pro's four tree clipboards. The tree clipboards are described in more detail at the end of this topic.

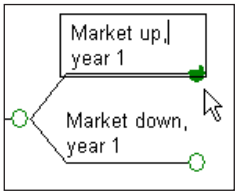
In addition to being able to duplicate and move subtrees in the tree window using the Edit menu commands described in this chapter, it is also possible to manipulate subtrees using the mouse and CONTROL key in the tree explorer view introduced in Chapter 2. See Chapter 21 for details on working in the tree explorer.

> ***Tip:***
> Besides text, tree nodes, and subtrees, other items that can be copied and pasted in TreeAge Pro include influence diagram nodes, variable definitions, and links with Excel. Items that can be copied from, but not pasted into, TreeAge Pro include text reports, model images, and graph images.

Endnodes

The Cut, Copy, and Clear Node commands will be available only if you have selected a single endnode (i.e., any node without branches, except the root node). Once a endnode has been cut or copied to the active tree clipboard, it can be pasted to any node.

► **To copy an endnode:**



▷ Select a node without branches.

▷ Choose **Edit >** Copy Node. Or, choose Edit > Copy… and select the Node option in the Copy Options dialog.

Once copied to the clipboard, the node can be pasted to any node as a new branch. It can be pasted to a node with or without branches, in the same tree or a different one. All aspects of the node, except for its branches, are included with the copy.

► **To paste a copied endnode:**

▷ Select any single node.

▷ Choose **Edit >** Paste Node. Or, choose Edit > Paste… and select the Node option in the Paste Options dialog.

After being pasted, the copied node remains on the active tree clipboard, and can be pasted at additional locations. It will remain on the clipboard until a different node (or subtree) is copied, or until TreeAge Pro is closed.

It is also possible to cut an endnode — i.e., remove it from its current location, in order to paste it in a new location.

► **To cut an endnode:**

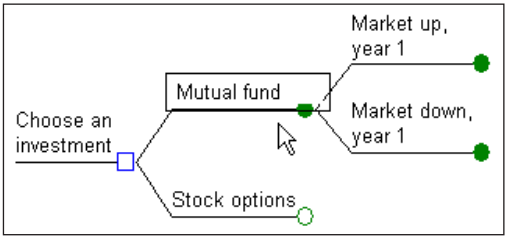
▷ Select a node without branches.

▷ Choose **Edit >** Cut Node. Or, choose Edit > Cut… and select the Node option in the Cut Options dialog.

Subtrees

Duplicating large tree structures, rather than building each one from scratch, can both save time and avoid errors. In many trees, some alternatives will be structural identical or closely similar, differing perhaps in particular probabilities or payoffs.

Before a subtree can be copied, it must be selected, as explained in Chapter 7.



► **To select a subtree:**

▷ Select a single node that has branches attached (i.e., not an endnode).

▷ Choose **Options >** Select Subtree, or control-click on the subtree's root node.

> ***Note:***
> Although the subtree's root node is highlighted along with the selected subtree, it will not be part of any subsequent operation on the subtree (i.e., the subtree root node will not be copied or cut).

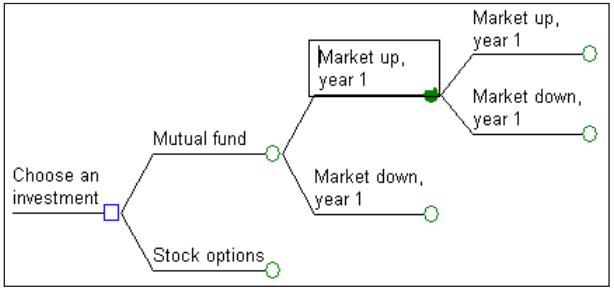
Once a node's subtree is selected, the Copy Subtree command becomes available.

► **To copy a subtree:**

▷ Select the subtree to be duplicated.

▷ Choose **Edit >** Copy Subtree (or Edit > Copy…, and select Subtree in the Copy Options dialog).

Once copied to the clipboard, the subtree can be pasted to any node — one with or without branches, in the same tree or a different one.

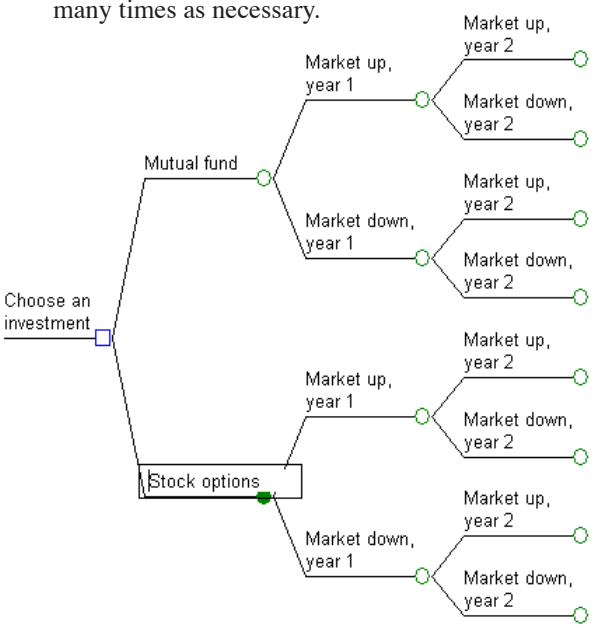


► **To paste a subtree:**

▷ Select any single node.

▷ Choose **Edit >** Paste Subtree (or Edit > Paste…, and select Subtree in the Paste Options dialog).

The Paste Subtree command can be repeated as many times as necessary.



Tip:

A copy of a subtree, while on the clipboard or after it is pasted, has no connection to the original subtree. See Chapter 21 for a discussion of *clones*, where copies of a subtree remain linked to the original, master subtree.

To move a subtree, rather than duplicate it, use the Cut Subtree commands.

► **To cut a subtree:**

▷ Select the subtree to remove (control-click on the parent node).

▷ Choose **Edit >** Cut Subtree.

This will cause the subtree (but not the subtree's root) to be removed from the tree and placed on the clipboard. As with a copied subtree, when a subtree is cut to the clipboard, the Paste Subtree command becomes available.

The Clear Subtree command is the only way to delete more than one node at a time. It remove the subtree without placing it on the clipboard.

► **To clear a subtree:**

▷ Select the subtree to be removed.

▷ Choose **Edit >** Clear Subtree.

Using the tree clipboards

TreeAge Pro has four tree clipboards, one of which is active at a time. This means, in effect, that you can cut or copy subtree X without losing subtree Y that is currently on the active tree clipboard. Prior to cutting or copying another subtree, simply activate one of the empty tree clipboard's by selecting it in the Edit menu.

The contents of the currently active tree clipboard can be viewed (but not modified) by choosing Edit > Show Clipboard. The Paste Subtree command pastes the contents of the currently active tree clipboard.

Cut, copy, paste, and clear text

It is possible to cut, copy, paste, and clear text in node descriptions, as well as in probabilities, payoffs, and any other formulas or values you type in TreeAge Pro.

Before cutting, copying, or clearing text, you must select the targeted letters, numbers, or words.

► **To select individual characters of text or formulas:**

▷ Either use the mouse to: A) click and drag from one end of the desired selection to the other; or B) click before one end of the selection, and then shift-click after the other end.

► **To select one or more words of text (or a formula):**

▷ Use the mouse to either: A) double-click on the first word of the selection and drag to the last word; or B) double-click on the first word of the selection, and then shift-double-click on the last word.

Once the text is selected, then you can choose the appropriate command from the Edit menu:

► **To copy (duplicate) selected text:**

▷ Choose **Edit >** Copy Text, or Edit > Copy….

▷ If the available menu command is Edit > Copy…, TreeAge Pro will open the Copy Options dialog box. In this case, click on the Text option, and press ENTER or click OK.

► **To cut (move) selected text:**

▷ Choose **Edit >** Cut Text. Or, choose Edit > Cut… and then select Text in the Cut Options dialog box.

The Copy and Cut commands both place the selected text on the clipboard. The text on the clipboard can then be pasted into any TreeAge document, or into another program.

> ***Tip:***
> The Cut, Copy, Paste, and Clear commands can also be accessed by right-clicking on the selected text and choosing the command from the quick menu.

The Paste Text command can be used to insert text or formulas both place the selected text on the clipboard. This text can then either be pasted into any TreeAge document, or into another program.

► **To paste (insert) text:**

▷ Place the text cursor in the desired location.

▷ Choose **Edit >** Paste Text. Or, choose Edit > Paste… and then select the Text option in the Paste Options dialog box.

The Edit > Clear Text command can be used to delete the selected text without placing it on the clipboard. The Clear Text command works identically to simply selecting text and pressing the DELETE key or the BACKSPACE key.

Undo and Redo

Each tree (and influence diagram) retains in memory details of the last actions that you took. Beginning with the most recent action and working back one action at a time, you can remove ***most all*** changes to structure, values, preferences, etc.

Note that the Undo "chain" is broken by some analysis and tree-building commands – TreeAge Pro will warn you also if a command cannot be undone.

The Undo item in the Edit menu reports what kind of change is the next one that can be undone. If the last action taken was to cut a node, the topmost command under the Edit menu will read Undo Cut Node.

► **To undo a change to your model:**

▷ Choose **Edit >** Undo [action].

If you go back too far in undoing modifications to the model, you can use the Redo command, also found in the Edit menu. The Redo command essentially allows you to undo previous Undo action(s).

► **To redo an undone change to your model:**

▷ Choose **Edit >** Redo Cut [action].

Reverting to a file's saved state

The Revert to Saved command clears all changes made to your model since it was last saved. Its effect is the same as simply closing the currently active tree (or other document) without saving changes, and then reopening the file.

> ***Warning!***
> The Revert to Saved command warrants careful use, as it cannot be undone.

► **To revert the current model to its saved state:**

▷ Choose **File >** Revert to Saved.

TreeAge Pro will prompt you to verify that you really want to clear all the changes you have made to the tree since it was last saved.

Find and replace text, formulas, and values

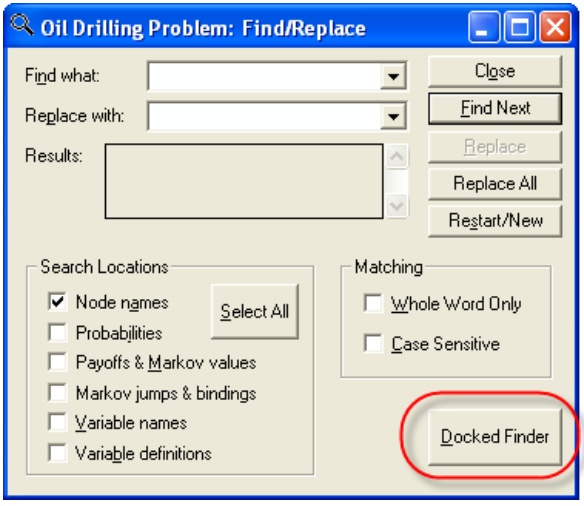
You can quickly search for and modify text and formulas at nodes in a tree using the Find/Replace tool. With some exceptions, this is often the best way to fix the spelling of a word used frequently in a tree, or to change a formula that appears in many payoffs, definitions, or other expressions. It is often useful also for simply finding particular nodes.

► **To find and/or replace text in a tree:**

▷ Choose **Options >** Find….

▷ Enter text in the Find What box, choose the desired Matching options, and select one or more search location(s).

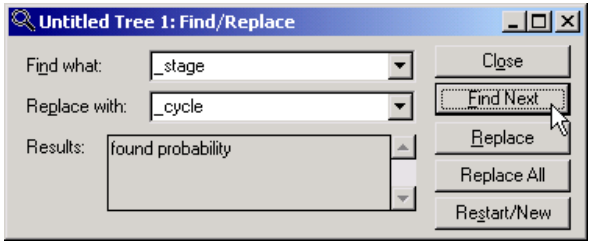
▷ Click the **Find Next** button to begin the search at the root node. The options groups at the bottom of the Find/Replace window will be temporarily hidden, to make it easier to see the tree window.



Docked Finder:

Instead of using the regular Find/Replace window, which will not remain visible if you click on the tree, you can switch to a docked Finder pane (click the **Docked Finder** button, or use the Display > Finder command).

If a match is found at a node, TreeAge Pro selects the node and reports the result. Clicking **Find Next** again will continue the search from where it left off. If text is entered in the Replace with box, clicking **Replace** will replace the current instance of the found text.



Clicking the **Replace All** button will repeat the Find Next and Replace operations until the tree is fully traversed.

Clicking the **Restart/New** button resets the search back to the root node, and allows you to change the search locations options.

If you check the Match whole word only option, TreeAge Pro will not search for partial word matches. If Match case is checked, matching text must have the same combination of upper and lower case letters as the search text that you specify.

To replace text in global formulas not included in any of the Find/Replace search locations (i.e., distribution parameters), try using the International Settings dialog, covered in the next section.

Tips:

- The Replace command is not recommended for changing a variable's name. Instead, edit the variable's properties in the Variables and Tables list (see Chapters 12–13).
- To select *all* nodes with a specific characteristic, such as a word in their name, try the **Options >** Select If… command, covered in Chapter 8.
- Changes made using Find/Replace cannot be undone, even while the dialog is open. After performing a replacement, reverting to an older version of the tree file may be the only way to reinstate modified text.

Using the Probability Wheel

A frequent problem encountered in decision analysis is the assignment of subjective probability assessments to chance events. Many people find it easier to use a graphical aid in assigning probabilities. One tool designed for this task is a probability wheel.

The Probability Wheel works whether or not you have already assigned probabilities to the branches. If you have assigned probabilities, they will be used as initial values for the wheel. The chance node can have up to seven branches; each branch will be assigned its own colored portion of the wheel.

As an example, the probability wheel could be used to aid in assigning relative weights to the two outcomes of the risky investment in the example tree used in earlier chapters.

► **To use the Probability Wheel:**



▷ Select a chance node with branches, and choose **Values >** Probability Wheel….

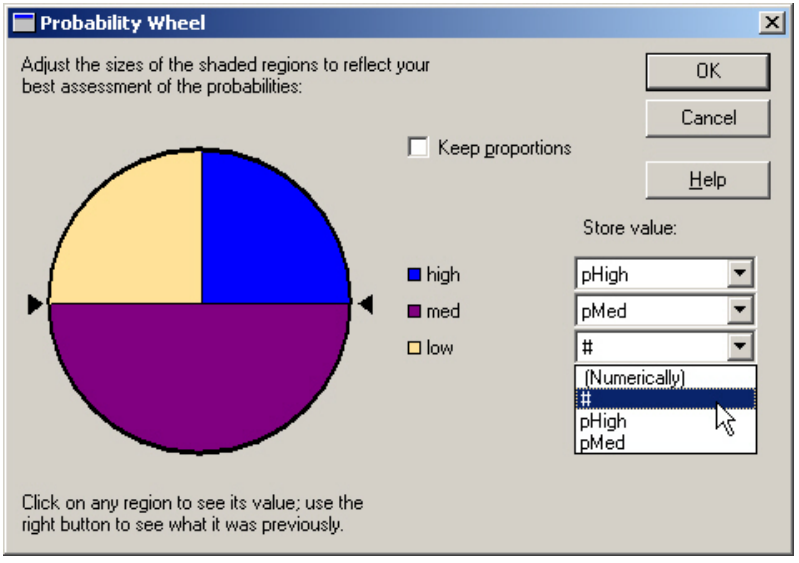
▷ Drag the pointers around the edge of the wheel until the sizes of the section match your best assessment of the relative likelihood of outcomes.

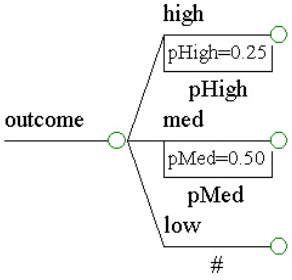
If the selected node has three or more branches, you will see a check box named Keep proportions. If selected, ratios are maintained on each side of the pointer you move.

Moving the mouse cursor over one of the pie wedges will change the pointer into a magnifying glass, indicating that you can click the mouse button to display the numeric value (probability) of the wedge. Right-clicking will display that wedge's starting value.

In addition to being able to store the probabilities numerically, in the branches' probability fields, variables can also be used (see Chapter 12). If you create variables before using the wheel, you can choose to store each wedge's value in one of three ways: numerically; as the "#" remainder probability; or in a variable defined at the branch. You may use # for only one branch.

The probability wheel is also available in the Define Variable window, allowing you to edit the value of a probability variable using the Wheel.





Chapter 10

Annotating the Tree

In addition to the basic ability to enter text labels at event nodes in the tree, TreeAge Pro includes a number of other options for annotating trees covered in this chapter.

The next chapter covers a variety of options for controlling the layout, formatting, and contents of decision trees.

In this chapter:

- Label nodes 116
- Node comments 117
- Note boxes and arrows 118

Label nodes

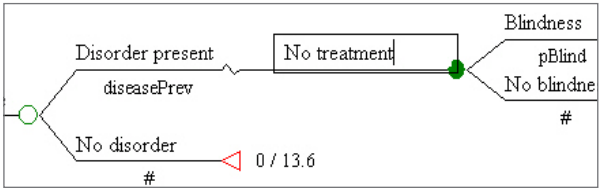
A label node, which uses a simple black "zigzag" as its symbol, acts like a placeholder. Label nodes have no impact on calculations, and cannot have more than one branch. A label node (or a series of them) can be inserted between event nodes to more clearly identify additional steps in a particular path.

► **To add a label node before an event node:**

▷ Select the event node and choose **Options** > Insert Branch….

▷ Select To Left, and click OK to create the new node.

▷ With the new node selected, choose **Options** > Change Node Type..., select the Label type, and click OK.



You could also change an existing endnode to a label node, and then use the Add Branch or Insert Branch command to add the one allowed branch to the label node. A node with more than one branch cannot be changed to a label node.

For calculation purposes, a label node behaves like a decision node with one branch, or a chance node with one branch having a probability of 1.0. The value of the label node is simply the value of the node immediately to its right.

Hidden node comments

Detailed comments can be assigned to the branches of a node and saved with the tree. Unlike note boxes (see below) and branch labels, the text of node comments is hidden.

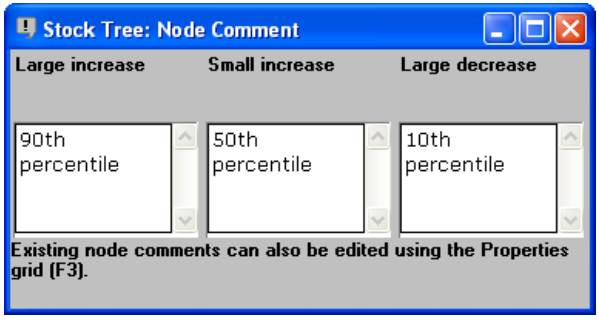
Node comments are particularly useful for recording the basis on which probability assignments were made for the branches of a chance node.

► **To add node comments:**

▷ Select a node that has at least one branch.



▷ Choose **Options >** Node Comment....



The window that appears will have one text area for each branch of the selected node, each labeled with the text description from the branch. You can move between text areas using the TAB key.

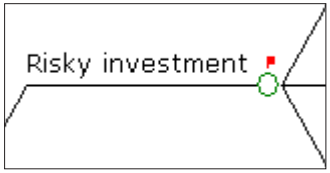
The open node comments can be printed. You may also print preview the node comments.

The font used to display and print the open node comments can be changed by choosing **Display >** Font.

► **To close the node comments:**

▷ Choose **File >** Close to return to the tree window (or click control box "X" button).

In the tree window, the node at which you entered a node comment is identified by a small red flag. This flag will not appear when the tree is printed, or when the Export or Copy Special commands are used to save or export a picture of the tree.



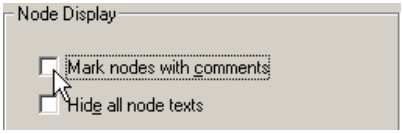
Double-clicking on a flag opens the Node Comment window again.

It is possible to turn off the screen display of flags.

► **To hide node comment flags:**

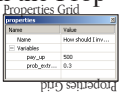
▷ Select **Edit >** Preferences... and choose the Node Display category.

▷ Select Mark nodes with comments, and press ENTER.



Editing comments in the Properties grid

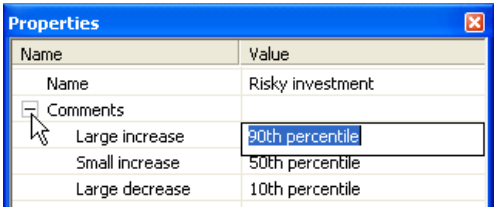
Once node comments are entered for the branches of a node, the comments can be edited in the Properties grid.



► **To edit comments in the Properties grid:**

▷ Select a node that has existing node comments, and activate the Properties grid under the Display menu.

▷ Expand the Comments group.



Note boxes and arrows

Models often benefit from the use of longer explanations than are desirable in branch descriptions. Node comments, described in the previous section, are useful for the model builder, but not for someone looking at a printout or image of the model.

Using note boxes and arrows, you can provide the model's audience with an overview of the whole model, or specific nodes or subtrees.

Creating note boxes

Any number of note boxes can be created. Boxes are initially placed independent of any node, but can be bound to particular nodes if desired. Each box's can use a different font, but all must use the same outline format (solid/dashed/none).

► **To draw a note in a tree:**

▷ Choose **Display >** Create Note (also available by right-clicking). The mouse cursor becomes a crosshairs.

▷ With the mouse, click and drag somewhere in the tree window. Make the box large enough to hold the text you wish to enter.



A rectangular box is created. In the upper-left corner of the box is a blinking text insertion caret, indicating that text can be entered in the note box by typing. Text entered in the note box will automatically wrap to the width of the box. Once the text box is full, it will not accept additional typed text until it is made larger (see the following section).

The font of the active note box can be changed. Each note box can use a different font, but all text within a single note box uses the same font.

► **To change the font of a note:**

▷ Click within the text in a note box.

▷ Select **Display >** Font....

▷ Change the font, size, or style, and click OK.

To see what the note box will look like when printed, unselect it.

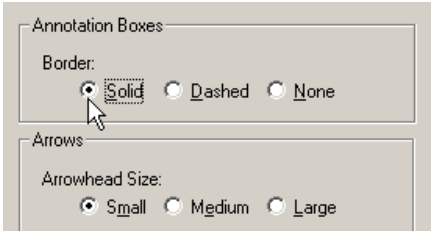
► **To unselect/deactivate a note box:**

▷ Click somewhere outside the box.

When a note box is inactive (i.e., not accepting text) and not selected, its outline is drawn on screen using dashed lines, by default. This is also what the note box will look like when printed. All note boxes in the tree use the same outline type — either dashed, solid, or none.

► **To change the tree's note box outline type:**

▷ Choose **Edit >** Preferences, and select the Notes & Arrow category.



This change will apply to all note boxes in the tree.

Moving, resizing, and deleting note boxes

In order to move, resize, or delete a note box, you must be able to select it. A note box that is activated for text entry, displaying the blinking text caret, is *not* selected.

► **To select a note box:**

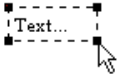
▷ Start with the mouse pointer inside the note box, showing the text cursor. Slowly move the mouse pointer towards any edge of the box, stopping as soon as the cursor changes to an arrow.

▷ With the arrow cursor, click on the box's edge with your mouse.

The selected note box will display a small square — a "handle" — at each of its four corners. These handles are used to resize the box.

► **To resize a note box:**

▷ Click on one of the box's handles, and click and drag it to change the size and proportions of the box.



► **To move a note box:**

▷ Click and drag on an edge of the note box, rather than a corner, and move the box to the desired location.

The Edit menu or the right-click menu can be used to clear, cut, or copy a note box.

► **To clear (delete) a note box:**

▷ Select the note box as described above.

▷ Choose **Edit >** Clear Note or Cut Note (or choose Cut Note from the right-click menu).

The Copy Notebox and Paste Notebox commands can be used to duplicate a selected note box.

Binding a note box to a node

When a note box is placed, it remains fixed in place relative to the top left corner of the tree window, and will not adapt to changes made to the tree. If nodes are added or deleted, for example, an existing note box may overlap with another object. This problem can be avoided by binding a note box to a node.

► **To bind a note box to a node:**

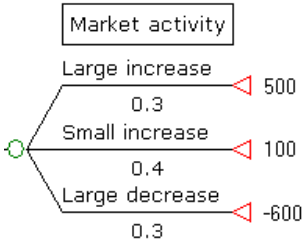
▷ Select the note by clicking on its outline.

▷ Select **Display >** Bind Note....

You will be prompted to select the node to which you want to bind your note.

▷ Click OK to continue, and select a node.

The bound note box will be aligned directly above the selected node, and will move with the node as the tree structure changes.



A bound note box cannot be deleted, cut, copied, moved or resized like an unbound note box. The note box will resize automatically to the length of the text. A bound note box will no longer wrap its text automatically. To wrap text, you must enter carriage returns in the text.

Tip:

Pressing the TAB key will move the text cursor from a bound note box to its node's branch line.

Creating arrows

You can draw arrows in conjunction with annotating a tree. Arrows might be used to associate an unbound note with one or more nodes.

► **To draw an arrow:**

▷ Choose **Display >** Create Arrow. The mouse cursor becomes a crosshairs.

▷ Click and drag with the mouse; start from the point where you want the arrow to begin, and release the mouse button where you want the arrow to point.

You can change the location of the arrow by clicking and dragging its line. It is possible to change the length and position of the arrow by clicking and dragging the endpoints of the arrow.

It is possible to customize the appearance of arrows in a tree, via the same Preferences dialog used to customize note box outlines (see above). Arrowheads can be small, medium, or large, and arrow lines can be solid, dashed, or dotted. Your changes will apply to all notes and arrows in the active tree.

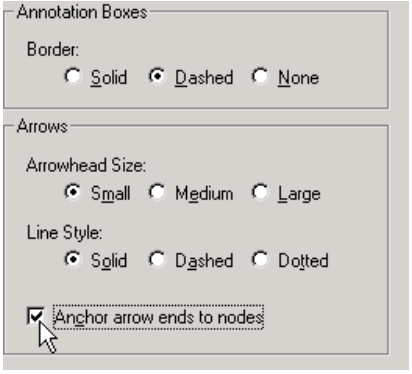
Tip:

To ensure that an arrow is drawn strictly horizontally or vertically, hold down the SHIFT key as you draw the arrow or change the location of one of its endpoints.

Binding arrows to nodes

By default, when an arrow is created it is not bound to any node. Therefore, structural changes made to the tree may cause nodes to move away from arrows. However, it is possible to have all arrows in a tree loosely bind to nearby nodes.

In the Arrows preferences, Check the option labeled Anchor arrow ends to nodes. If this setting is active, and both the tip and end of an arrow are associated with different nodes, then the arrow will actually stretch when structural changes to the tree require it.



Chapter 11

Tree Display Preferences and Options

This chapter covers a wide variety of options for controlling the layout, formatting, and appearance of trees. Some of these features apply to calculated values displayed following roll back, while others affect display of the tree's structure.

In this chapter:

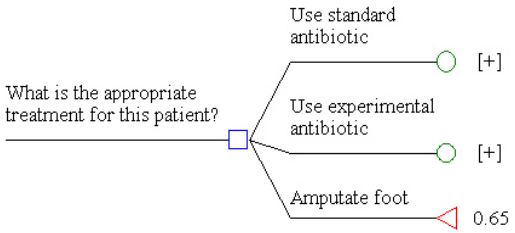
- Collapsing subtrees 122
- Aligning selected nodes 123
- Displaying roll back columns 124
- Other tree display preferences 128
- Changing fonts 132

Collapsing/hiding subtrees

The Collapse Subtree command can be used to temporarily hide any subtree (including the root node's subtree). It is particularly helpful when working with or presenting large trees.

► **To collapse a subtree:**

▷ Select a node with visible branches, and choose **Display >** Collapse Subtree.



The subtree emanating from the selected node is hidden and a plus sign (+) is displayed in its place, to the right of the node. The plus sign will also appear in place of the hidden subtree in printouts and exported images of the tree. Collapsing a subtree does not affect calculations.

Expanding hidden subtrees

Hidden subtrees can be uncollapsed in two ways: showing one generation of branches; or unhiding the entire subtree.

► **To expand a collapsed subtree one generation at a time:**

▷ Select a node with a plus sign to the right, and choose **Display >** Expand Subtree Once.

The branches of the selected node will be displayed, but any subtrees attached to these branches will remain hidden and plus signs displayed in their place.

► **To expand an entire collapsed subtree:**

▷ Select a node with a plus sign to the right, and choose **Display >** Expand Entire Subtree.

Notes:

- The CONTROL + 0 (zero) keyboard shortcut will automatically move the selection one node to the left and then collapse that node's subtree. The other CONTROL + *number* keyboard shortcuts will automatically uncollapse, if needed, and move the selection to the numbered branch to the right.
- The Collapse or Expand Subtree commands can be used on more than one node at a time. If you select multiple nodes having branches, the collapse command will be available; if you select multiple nodes showing plus signs where subtrees have been collapsed, the expand commands will be available.
- Visible clone copies (see Chapter 21) can be collapsed.
- Collapsing a subtree in the tree explorer pane does not collapse the subtree in the regular tree view.

Aligning selected nodes

TreeAge Pro includes two options for adjusting the vertical alignment of nodes in a tree. This section deals with the use of the Skip Generation command to make manual adjustments to the vertical alignment of selected nodes. It is also possible to specify that all terminal nodes be vertically aligned automatically; this is covered in a later section, on tree display preferences.

In many trees, asymmetry in intervening events results in the related nodes not lining up vertically. It is sometimes desirable to force these related nodes to line up, resulting in a more intuitive layout of the tree.

► **To align a selected node with a node in a different subtree:**

▷ Select a node.

▷ Choose **Display >** Skip Generation, to move the node's subtree one generation to the right.

Skipping one or more generations can be used to line nodes up vertically — for example, the multiple nodes representing a particular decision, in different paths. To skip more than one generation, simply select the menu command repeatedly.

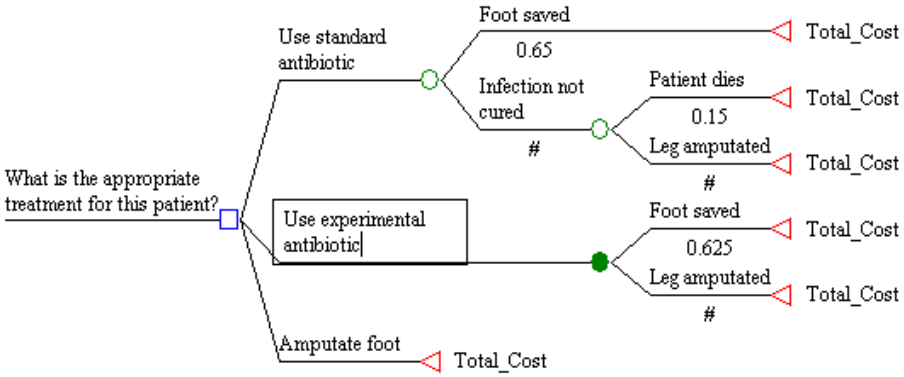
To reverse the effects on the tree, simply un-skip generations at the node which skips a generation.

► **To remove an extra generation from the selected node:**

▷ Select a node which is set to skip a generation.

▷ Choose **Display >** Unskip Generation, to move the subtree to the left.

Skipping generations does not affect calculations.



Displaying roll back columns

When a tree is rolled back, TreeAge Pro normally displays next to each terminal node a roll back box containing the node's calculated payoff and, in an optimal path, its path probability. Chapters 5 and 7 described some of the customizations that can be made to the roll back display.

Instead of showing the standard roll back boxes at end nodes, a tree can be set up to display user-defined columns of values to the right of visual end nodes during roll back.

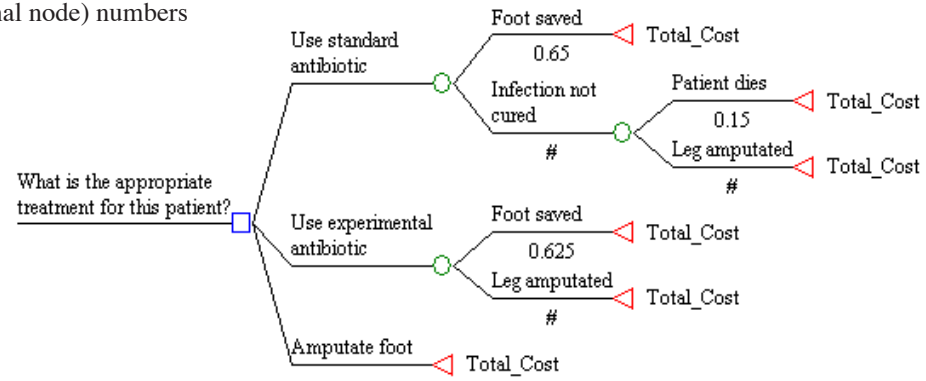
Calculated values and other information that can be displayed in roll back columns include:

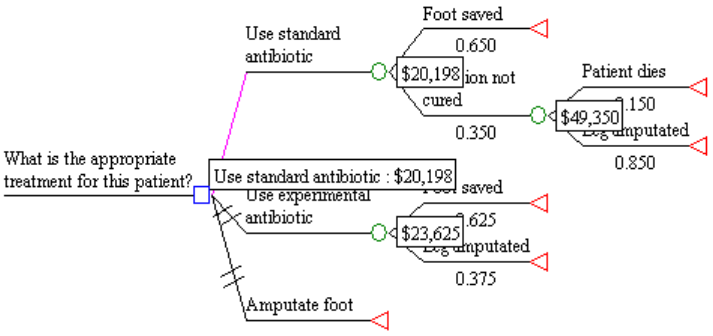
- payoffs, including extra (non-active) payoffs
- individual components of a complex payoff formula
- path probabilities
- scenario (i.e., terminal node) numbers

Some important features of roll back columns are:

- During roll back, the table of values can be copied to a spreadsheet or other application for reporting or further analysis.
- A row will be displayed for every *visual* end node, even those that are not terminal nodes — for example, if a node's subtree is collapsed (see above) or is a hidden *clone* copy (see Chapter 21), a row is shown for the node.
- Calculated values can be displayed using custom numeric formatting.

The example model used in this section, shown below, is the Terminal Columns tree, found in TreeAge Pro's Tutorial Examples subdirectory. The tree is shown with roll back columns at the bottom of the page.

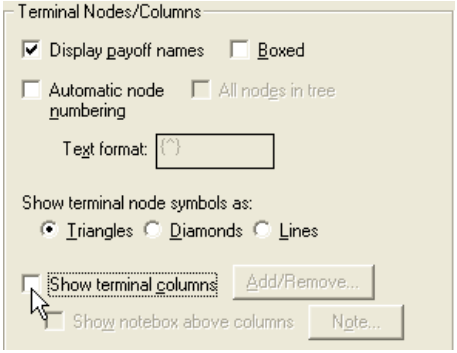




	Cost	Var: Total_Cost	Path Prob	Contribution
	$4,500	$4,500	0.650	$2,925
	$14,500	$14,500	0.052	$761
	$55,500	$55,500	0.297	$16,511
	$4,500	$4,500		$2,813
	$55,500	$55,500		$20,813
	$24,000	$24,000		$24,000

► **To turn on the display of roll back columns:**

▷ Choose **Edit >** Preferences…, and select the category labeled Terminal Nodes/Columns.



▷ Check the Show columns option, and click the **Add/Remove…** button.

The Columns dialog is displayed, which is used to create and manage the list of roll back columns.

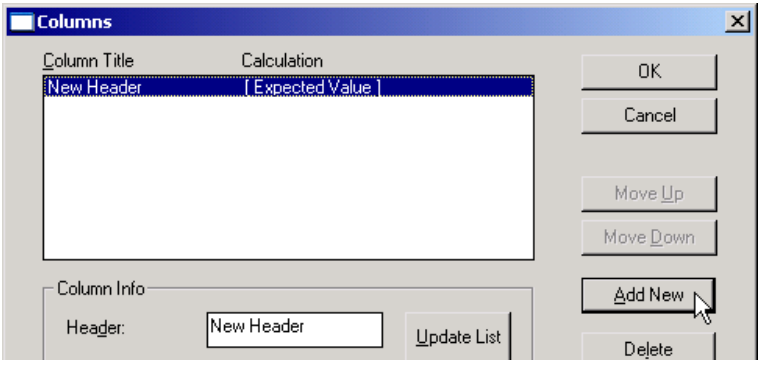
► **To create a column:**

▷ Click the Add New button

▷ Enter a short title for a new column in the Header field, and then choose the appropriate calculation and formatting.

▷ The new column will automatically be updated when you click Add New again, or close the dialog.

▷ Click Update List to show the new column properties in the list.



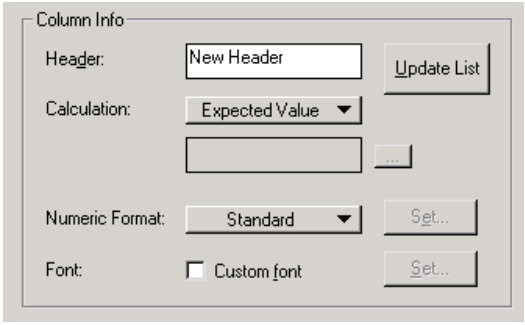
Each column has these properties:

Header – Title to be displayed, underlined, at the top of the column.

Calculation – Value displayed at each visual end node.

Numeric Format – Use either the numeric format from the tree, a custom numeric format, or no formatting.

Font – Specify a custom font, or default to the tree's "EV" font.



With the display of columns turned on in the Preferences dialog, value columns will be displayed when the tree is rolled back, with rows for each visual end node, unless the value cannot be calculated.

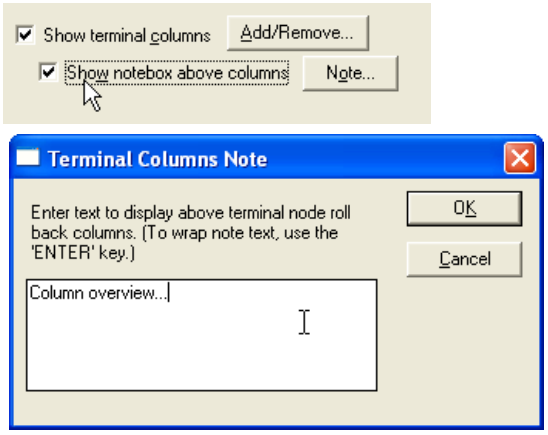
The display of roll back columns can be turned off in the Terminal Node/Columns preferences, without deleting the list of columns.

► **To turn off the display of roll back columns in a tree:**

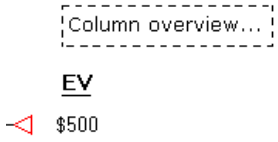
▷ Choose **Edit >** Preferences…, and the Terminal Nodes/Columns category.

▷ Uncheck the Show columns option. Click OK.

To annotate the columns, use the optional notebox that can be displayed above the columns during roll back (new setting in v2006 release 1.x).



The text in the roll back column notebox cannot be edited from within the tree window – use the **Note…** button in the Preferences dialog.



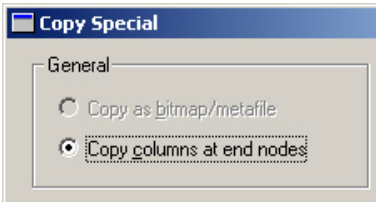
Copying columns to Excel

If roll back does not provide a calculation that you are interested in (a common one is the contribution each end node makes to the total expected value), it may be possible to do the calculation in Excel using the output from roll back columns. It is possible to copy the table of roll back columns as tab-delimited text, and paste it into a spreadsheet.

► **To copy roll back columns to the clipboard:**

▷ Roll back the tree, then choose **Edit >** Copy Special….

▷ Select the option labeled Copy columns at end nodes.



▷ Switch to the target document and paste the text.

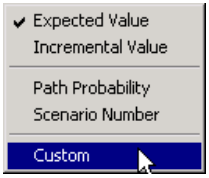
The names of the end nodes are included as the first column in the table.

Incremental calculations at decision nodes

If an incremental-type calculation column is created, values will be displayed in the column only if the branches of a decision are visual end nodes. Incremental calculations will only be performed between the branches of a decision node.

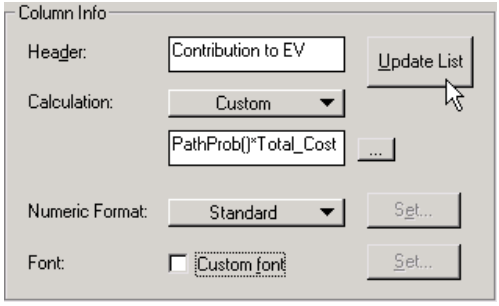
Assuming the branches of your decision node are not terminal nodes, then you must use the Collapse Subtree command (see above) to make the decision node's branches visual end nodes, by hiding their subtrees temporarily.

Custom roll back calculations

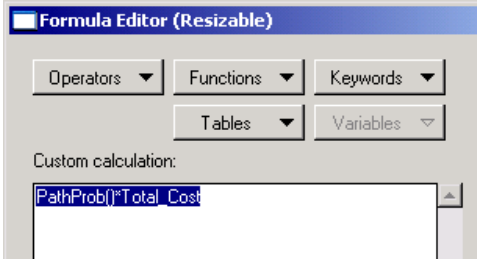


If you choose the Custom calculation type, you must enter an expression to calculate at each end node.

A variable or formula can be directly entered into the text box below the Calculation drop-down list.



Clicking on the ellipses button (…) to the right will open the Formula Editor dialog, with tools for building the expression. (The use of variables and formulas is covered starting in Chapter 13.)



As noted in Chapter 7, in the section on the Calculate extra payoffs preference, it is possible to use a roll back column to report calculated values for one or more of extra (non-active) payoffs.

If the active payoff is #1, and you set the calculation preferences to calculate the extra payoffs #2, a custom roll back column could report the calculation "Node(*2*)". Collapsing the tree back to the decision node branches would display the expected values for extra payoff.

Roll back column tips:

- To suppress the display of roll back boxes at internal nodes, so that only roll back columns display during roll back, go to the Roll Back category of the Preferences dialog and select Roll back calculates payoffs only.
- If your columns do not display properly, check to see if the minimize empty space setting in the Tree Display page of the Preferences dialog is selected. If so, turn it off. If roll back columns still do not display correctly, try choosing **Display >** Redraw.
- To show terminal node (scenario) numbers in a column, you must activate the automatic node numbering setting, found in the Terminal Node category of the Preferences dialog (see below).

Other tree display preferences

Each tree created in TreeAge Pro can be given its own distinct set of display preferences. See Appendix C for settings not covered here.

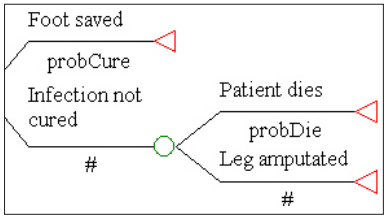
Hiding and boxing payoffs

By default, the active payoff formula is displayed at each terminal node when the tree is not rolled back. This information can be hidden if, for example, you want to simplify a visual presentation of the tree.

► **To hide terminal node payoff formulas:**

▷ Choose **Edit >** Preferences…, and the Terminal Nodes/Columns category.

▷ Uncheck the option labeled Display payoff names, and press ENTER or click OK.



Or, the tree's payoffs can be displayed in boxes, even when the tree is not rolled back.

► **To enclose payoffs in boxes when the tree is not rolled back:**

▷ In the Terminal Nodes/Columns preferences category, select the option labeled Display payoff names, and the option labeled Boxed.

Hiding probabilities and branch labels

The numbers or formulas entered for probabilities can be hidden from view. Node branch labels can also be hidden.

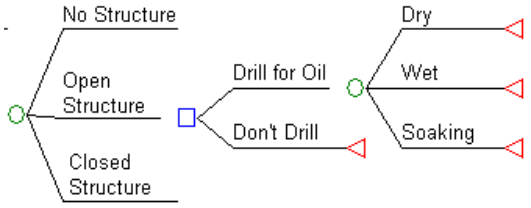
► **To turn off the display of probabilities and/or node names:**

▷ Choose **Edit >** Preferences, and the Node Display category.

▷ Check the option labeled Hide probabilities only, or the option labeled Hide all node texts. Press ENTER or click OK.

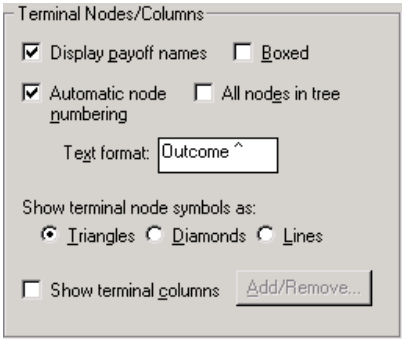
Displaying a "skeleton" tree

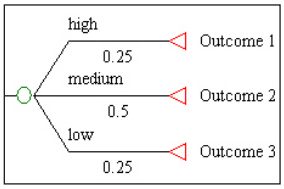
TreeAge Pro includes a "skeleton" tree setting that displays the tree in an compact format useful for display or printing. This setting separate each node from its branches (except for the root node).



Also, collapsed subtrees are completely hidden (no plus signs). The skeleton display format is best used in combination with the Minimize empty space, Hide node probabilities, and Hide payoffs preferences (e.g., see the Skeleton Tree file, shown here, in the Tutorial Examples subdirectory).

Terminal node numbers





Terminal nodes can be automatically numbered, and these numbers displayed to the right of terminal nodes (or in a roll back column).

► **To show scenario/terminal node numbers:**

▷ Open the Terminal Nodes/Columns preferences category.

▷ Check the option labeled Automatic node numbering

▷ Enter the text to be used for node numbering (see below). Press ENTER or click OK.

The text you enter for terminal node numbering must use the caret ("^") as a placeholder for the node number. The caret can be used alone or with additional text, as in `Outcome ^`. The terminal node number and text will be displayed whether or not the tree is rolled back.

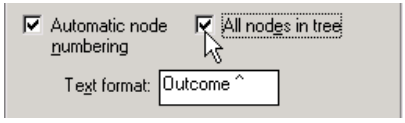
Number all nodes

Instead of just numbering terminal nodes, TreeAge Pro can apply the numbering format you specify to **all** nodes in the tree.

► **To show node numbers for all nodes:**

▷ In the Terminal Nodes/Columns preferences, check the option labeled Automatic node numbering.

▷ Also check the sub-option All nodes in tree.



▷ Enter the text to be used for node numbering. Press ENTER or click OK.

Increasing or decreasing vertical white space

If two branches that are vertical neighbors appear too close together, there is the way to increase the space between them.

► **To increase the vertical spacing between two nodes:**

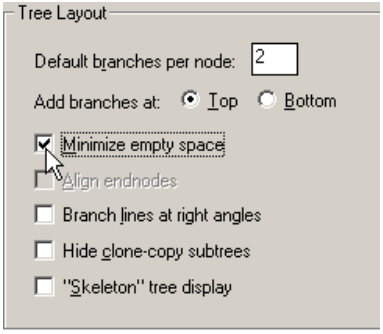
▷ Click on the bottom branch's text label, to the left of the first word in the label. Press ENTER one or more times, inserting carriage returns until you have created sufficient white space

A tree can be compressed vertically, reducing white space and yielding a very compact tree.

► **To compress a tree vertically:**

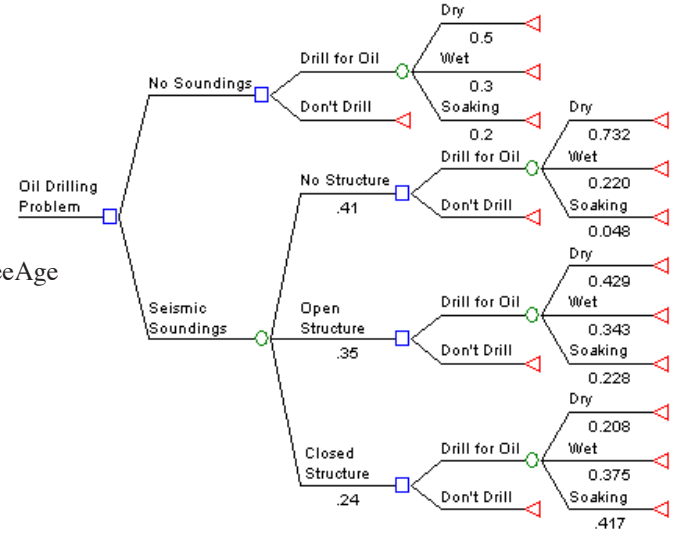
▷ Choose **Edit >** Preferences…, and the Tree Display category.

▷ Check the option labeled Minimize empty space. You must first ensure that Align endnodes is *not* selected.



▷ Press ENTER or click OK.

The graphic shown at right illustrates how TreeAge Pro compresses the display of the tree.



If the display of your tree looks broken when the Minimize empty space setting is turned on, review these potential conflicts:

- The Minimize empty space and Align endnodes settings are incompatible.
- Minimize empty space can cause problems when a tree is rolled back, because of a lack of space for roll back boxes.
- Using both the Minimize empty space and Branch lines at right angles settings can result in branch lines which slice through node symbols.
- Minimize empty space is likely to cause problems with the display of terminal node columns.

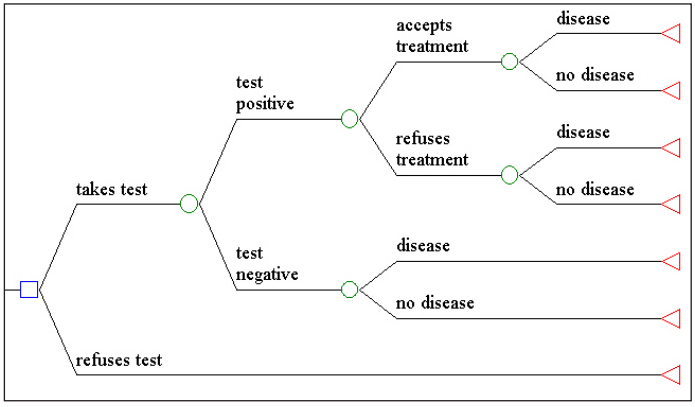
Vertically aligning terminal nodes

There is a quick way to align all terminal nodes in a tree with the right-most terminal node.

► **To align all terminal nodes at the right edge of the tree:**

▷ Choose **Edit >** Preferences, and select the Tree Display category.

▷ Check the option labeled Align endnodes. Press ENTER or click OK.



For instructions on aligning specific nodes in the tree, see the section at the beginning of this chapter.

Changing fonts

The font, size, and style of any text that appears in a tree can be easily changed. Fonts can either be changed for selected objects or globally, via a tree's preferences.

Changing the font of selected objects

One way that the appearance of text entered on the face of a document is by selecting a node, subtree, or note box.

► **To change the font of the selected node(s) or note box:**

▷ Select a single node or note box.



▷ Choose **Display >** Font... (or click the Font tool bar button).

▷ Change the font, size, or style, and click OK.

If a node is selected, this will only change the font of the node description, on the branch to the left. If the selected node is a branch of a chance node, the font used for the probability is modified separately.

► **To change the font of the selected node's probability:**

▷ Select a branch of a chance node and tab into its probability.

▷ Select **Display >** Font...

▷ Change the font, size, or style, and click OK.

Note on node fonts:

Once a selected node's font has been modified, it is independent of the tree's global font preferences, described on the next page. Changes made to the tree's font preferences will not affect the selected node.

When a selected subtree's font is modified, both the node descriptions and probabilities are affected.

► **To change the font of an entire subtree:**

▷ Hold down the CTRL key and click on a node that has branches; this will select the node's subtree.

▷ Select **Display >** Font....

▷ A dialog box appears asking you to confirm that you wish to change the font of the entire subtree. Click OK.

▷ Change the font, size, or style, and click OK.

Unlike most subtree operations, changing the font of the subtree includes the subtree's root node in the operation. The root node, and all of its descendants, will use the font specified for displaying node text as well as probabilities.

Changing font preferences

The Fonts page of the Preferences dialog makes it possible to specify, for the entire tree, the font used for node names, probabilities, expected value boxes, and (if displayed) definitions of variables. Each button calls up the standard font, size, and style dialog, but changes made in those dialogs apply only in the limited context that their names reflect.



The Node Font button will change the branch text font for nodes subsequently created in the tree. It is also applied to existing nodes in the active tree, with the exception of selected nodes and subtrees at which you have individually changed the font.



The Prob Font button will change the probabilities font for the active tree. Like the Node Font, this is used for new nodes and existing nodes in the active tree, with the exception of nodes and subtrees at which you have individually changed the probability field font.



The EV Font button allows you to change the font for roll back boxes displayed upon roll back of the active tree.



The Variables Font button allows you to change the font for variables displayed beneath nodes in the active tree, if you elect to display full variable definitions in the tree (see Chapter 13).

Like other preferences, font settings can be saved as defaults for subsequent trees. Simply check the box entitled Save settings as default (on the right-hand side of the Preferences dialog) and click OK. When you close the Preference dialog, the active tree's preferences will be saved on the computer and will govern the font settings of new trees that you create. (Note that *all* preferences will be saved, not just those in the Fonts page.)

Uncertainty and Variability: Using Variables, Tables, and Distributions

The chapters in Part III focus primarily on representing uncertain parameters using variables and distributions, and performing sensitivity analysis and Monte Carlo simulation.

The last chapter in Part III provides information on storing analyses that you may need to repeat frequently and that require a complicated setup procedure (like sensitivity analysis and Monte Carlo simulation).

In this part:

- Chapter 12, "Introduction to Variables and Sensitivity Analysis"
- Chapter 13, "Working with Variables"
- Chapter 14, "Building Formulas Using Variables and Functions"
- Chapter 15, "More Sensitivity Analysis Tools"
- Chapter 16, "Using Distributions and Monte Carlo Simulation"
- Chapter 17, "Distribution Functions, Options, and Types"
- Chapter 18, "Creating and Using Tables"

Chapter 12

Introduction to Variables and Sensitivity Analysis

This chapter provides a basic tutorial on the use of variables and one-way sensitivity analysis in decision trees.

Subsequent chapters cover TreeAge Pro's many useful tools for working with variables, as well as more complex, multi-way and probabilistic sensitivity analysis. In particular, refer to the first sections of both Chapter 14 (on defining variables non-numerically) and Chapter 15 (on avoiding sensitivity analysis problems).

Some users may wish to refer to the Index to find out more about special uses of variables in Markov microsimulation, user-defined Python functions, and clones.

In this chapter:

- Sensitivity analysis background 138
- Using variables in a tree 139
- Performing one-way sensitivity analysis 148
- Sensitivity analysis thresholds 151

Sensitivity analysis background

Sensitivity analysis was introduced in Chapter 1 as a means of assessing the extent to which a model's calculations and recommendations are affected by uncertainty. Specific questions about the model that sensitivity analysis can help answer are:

1) Is a model ***sensitive*** to a particular uncertainty — e.g., does varying a parameter's value result in changes in optimal strategy?

2) If a model is sensitive to a particular uncertainty, at what value(s) of the parameter does the model recommend a change in strategy?

3) Does the sensitivity analysis result make sense? (This is a model debugging question.)

The tutorial in this chapter shows how to prepare a tree for sensitivity analysis, how to perform a one-way sensitivity analysis, and how to interpret the results. Chapter 15 covers multi-way sensitivity analysis and tornado diagrams. Chapter 16 covers the use of probability distributions and Monte Carlo simulation to analyze models with complex or numerous uncertainties.

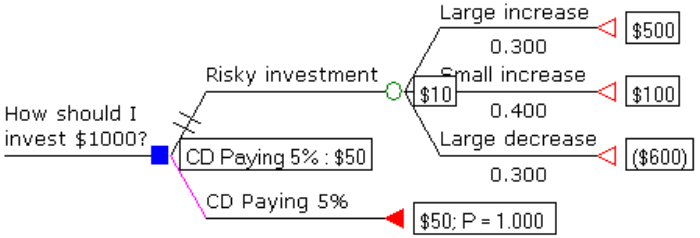
Chapters 13 and 14 provide important details on working efficiently with variables in decision trees. The information in these chapters can help you improve your productivity when building complex decision trees, and also insure against costly modeling errors.

Variables and sensitivity analysis

Up to this point in the tutorial, decision trees have been analyzed on the basis of baseline, numeric values for payoffs and probability values. In order to perform sensitivity analysis on an uncertain quantity, however, its numeric value must be replaced with a ***variable*** — a named parameter.

In the investment decision modeled in Chapters 1—3, subjective estimates for probabilities and payoffs were used. The resulting tree is shown below, rolled back. The expected value calculations suggest that the CD investment is optimal.

The basic uncertainty is that, at the time of the decision, the investor cannot know what the price of the equity will be in one year. A simple probability distribution — a chance node with three branches — represents a range of possible changes in the risky investment's value. Assuming that you have a mix of different expert opinions about the risky investment, it would be useful to be able to perform sensitivity analysis on the related parameter uncertainties. The extreme, 10th and 90th percentile values in the payoff distribution — a $600 decline or $500 rise in value — are possible candidates for sensitivity analysis. Another option is to vary the probabilities.



The next section of this chapter shows how to replace selected numeric values in the tree with variables. **This is a prerequisite to performing sensitivity analysis in TreeAge Pro.**

Using variables in a tree

In TreeAge Pro, a variable is a named parameter which functions as a placeholder for a numeric value (or a formula). Variables have a variety of functions in TreeAge Pro, including:

- representing uncertain or unknown values, usually in preparation for sensitivity analysis
- acting as placeholders for mathematical formulas that include functions, tables, spreadsheet links, and other variables, for example in defining a complex payoff

The tutorial in this chapter focuses on the use of variables as a basis for sensitivity analysis. Chapter 14 will discuss the creation of payoff formulas using variables.

When building complex models, variables are often used from the outset. The investment decision tree has already been completed using numeric payoffs and probabilities, however, so this tutorial will take a different course, replacing existing numeric values with variables.

> ***Tutorial Notes:***
> Some procedures explained in this chapter are not required steps for using variables in the investment tree. Lines starting with the ⇨ character are required tutorial steps; other instructions start with the ▷ character.

Steps for using variables

There are three basic steps to remember when using a variable as a parameter in a decision tree:

1) ***Declare name*** — Based on its intended function in your model, decide on a clear name for the variable (following the naming guidelines outlined below). Add the name to the list of recognized variables in the tree.
2) ***Define, assign*** — Define the variable at a node, often the root, by assigning it a value (or a formula).
3) ***Use*** — Anywhere the corresponding value is used in the tree (e.g., payoffs or probabilities), in its place substitute the variable name.

TreeAge Pro offers multiple methods for each step; this chapter illustrates a few possible methods.

Guidelines for naming variables

Variable names must conform to certain rules (similar to Microsoft® Excel's rules for cell names). Each variable name must:

- begin with a letter or underscore character ("_")
- contain only letters, numbers, and underscore characters
- be no longer than 32 characters

TreeAge Pro will alert you if try to use an invalid variable name.

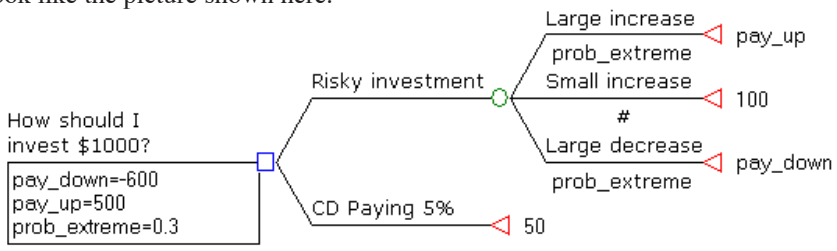
It is recommended that you follow some kind of naming convention when creating variables in a particular model. For example, you could use the prefix "prob" or "p_" for probability variables, "c" for cost variables, and so on.

Variable names are not case-sensitive. For example, the names probUp, PROBUP and probup are equivalent; prob_UP would be a different variable, however, since it includes an extra character.

Creating and defining variables

As explained at the beginning of this chapter, there are two apparent sensitivity analysis approaches in the investment tree: vary the extreme probabilities or the extreme payoff values.

The tutorial will start by using variables in the branch probabilities of the extreme outcomes, and after that in the payoffs. The tree will eventually look like the picture shown here.

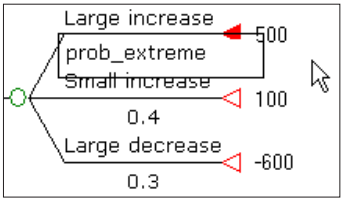


It is a good idea, if you already have a working version of your tree without any variables, to keep a backup copy of the tree.

► **Prepare a copy of your tree:**

- ⇨ Open the Stock Tree created in Chapter 2. Or, locate the copy of this file included in the Tutorial Examples_backup\ subdirectory of the TreeAge Pro program directory.
- ⇨ Save a copy of the file. We'll call it the Three Variables tree.
- ⇨ Changes cannot be made to a tree's values or structure while it is rolled back. If the tree is currently rolled back, choose **Analysis >** Roll Back to turn off roll back and allow variables to be added to the tree.

The first step is to replace the numeric probabilities of the Large increase and Large decrease branches with a variable placeholder. One method of adding a variable to the tree is to simply type a new variable name into a probability, payoff, or other expression.



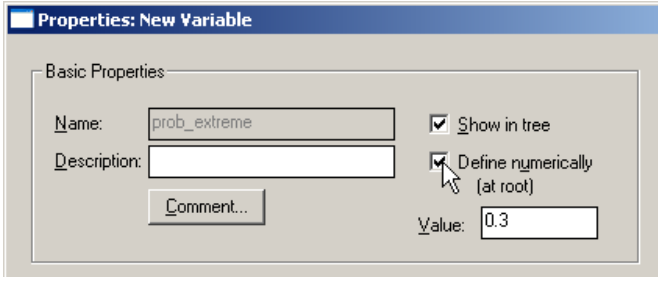
► **Defining a new variable: (Method 1)**

⇨ Select the Large increase terminal node and press the TAB key to edit the probability. Or, simply click with the mouse in the probability editor, below the branch line.

⇨ Delete the numeric probability of 0.3, and in its place type a new variable name `prob_extreme`.

⇨ Press TAB (or click outside the node) to have TreeAge Pro check the changes you made to the probability expression.

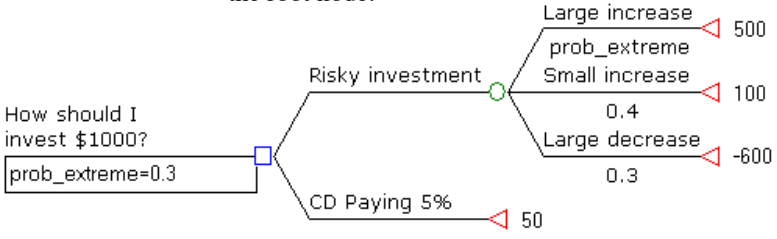
TreeAge Pro does not recognize the name you entered in the probability editor, and will assume that you want to create a new variable. A Properties dialog will be displayed for the new variable, in which you can describe and numerically define this parameter.



⇨ Check the box labeled **Define numerically (at root)**, and enter the baseline value of 0.3 (or 30%) in the Value box. You can enter a description and sensitivity analysis range later.

⇨ Press ENTER or click OK to save the variable properties and return to the tree. (Or, press CANCEL if the name was typed incorrectly.)

The name "prob_extreme" has been added to the list of recognized variables names in the tree, and it has been assigned a default numeric value of 0.3 at the root node.



The assignment of a value (or formula) to a variable at a node is called a *definition*. Which node you define a variable at determines where the variable definition will apply in the tree (i.e., which payoffs and probabilities can use the variable name).

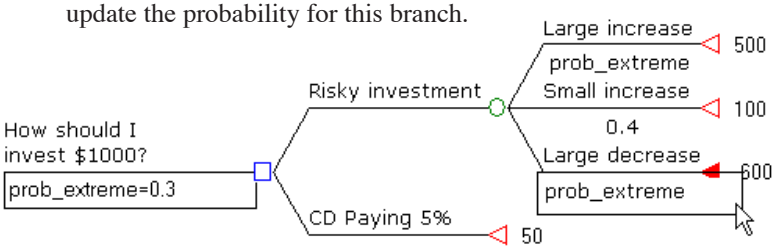
The root node definition of prob_extreme, for example, applies at the root node *and everywhere to the right*, including at the Large increase terminal node, where the variable is already referenced, and at the Large decrease node, which must now be updated to use the variable.

► **To insert an existing variable name:**

⇨ Make the same change to the Large decrease terminal node. Delete its numeric probability of 0.3.

⇨ Instead of typing the variable name prob_extreme, you can choose **Values** > Insert Variable…(or press CTRL+i). TreeAge Pro will either display a list of variables to choose from, or automatically insert the variable that matches a partial name that you have entered.

⇨ Press TAB (or click outside the node) to update the probability for this branch.



In this case, TreeAge Pro does not prompt you to create the variable prob_extreme, since the tree already includes a variable with this name.

Tip:

A common method of setting up variables which is not illustrated in this chapter is to set up the entire list of variable names *before* creating the structure of the tree. This can be done using the Variables and Tables dialog, described in detail in Chapter 13.

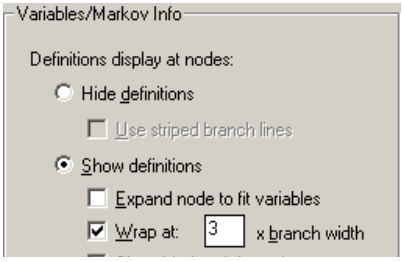
Finding and fixing problems with definitions

The variable definition, prob_extreme = 0.3, should be visible at the root node of the tree. If the definition is ***not*** visible below the root node, as in the above picture, it does not mean that the definition does not exist. The first thing to check is whether the display of variable definitions has been turned on in the tree's preferences.

► **To display variable definitions in the tree:**

▷ Select **Edit >** Preferences…, and choose the Variables Display category from the list.

▷ Check the option labeled Show definition, and also Expand node… (or Wrap at…). Ignore the other settings for now.



▷ Press ENTER or click OK to return to the tree.

If the tree is set to display variable definitions, but the definition prob_extreme=0.3 does not appear at the root node, determine whether the variable was created (i.e., added to the tree's list of recognized names).

The list of named variables in the tree can be viewed and modified in a number of ways. A quick way to access the variables list is by ***right-clicking*** on a node (use the root node for now).

► **To view the list of tree variables in a "quick" menu:**

▷ Right-click on the root node and choose Define Variable.

▷ If any variables exist in the tree, their names will be listed (below the New… command).



Later, we will look at using the Variables and Tables dialog to manage a tree's variables.

► **To open the Variable and Tables list:**

▷ Choose **Values >** Variables and Tables….

If the variable is not listed, choose the New… command to add the variable to the tree and define it at the selected node. If the name already appears in the list, select its name in the list and define it at the selected node.

If the Properties grid is visible, it can be used to either modify or delete a definition at the selected node. Simply expand the list of variables defined at that node, and click in the box showing the current definition.



Adjusting complementary probabilities

Our changes to the probabilities in Three Variables tree are not complete. If the value of the variable prob_extreme is changed, for example during sensitivity analysis, the probabilities at the chance node will no longer sum to 1.0 — unless a change is made to the 0.4 probability at Small increase.

The two options for Small increase's probability are to:

- enter an expression in terms of the new variable (e.g., "1-prob_extreme*2"); or
- use TreeAge Pro's automatic complement calculator ("#').

Let's use the second option, the #.

► **To assign a remainder expression to a probability:**

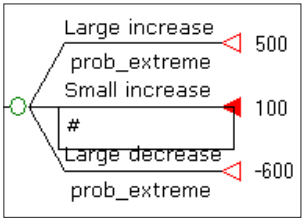
▷ Delete the numeric 0.4 probability of the Small increase node.

▷ Replace it with # (a hashmark), which will still calculate as 0.4 unless the definition of prob_extreme changes.

Now, if we change the value of prob_extreme to 0.25, the probability of Small increase will automatically be recalculated as 1 - 0.25 - 0.25, or 0.5, and the chance node's probabilities will continue to be coherent (sum to 1.0).

Once prob_extreme has been defined and used properly, and the chance node's probabilities have been adjusted appropriately, you could skip ahead to the section on performing sensitivity analysis, and try analyzing the impact of this uncertainty on your decision.

The tutorial continues at this point, however, by illustrating more ways of creating variables — in this case, to represent uncertain payoffs.



Creating and defining variables before using them

Using variables in payoffs instead of just numbers is very similar to using variables in probabilities. Using variables allows components of payoffs to be tested using sensitivity analysis, and also updated more easily.

Chapter 15 will show how to create more complex payoff formulas, in order to separate the payoff calculation into components (like principal, interest rate, and rate of return). In the current exercise, however, simple payoff variables will be used, rather than formulas.

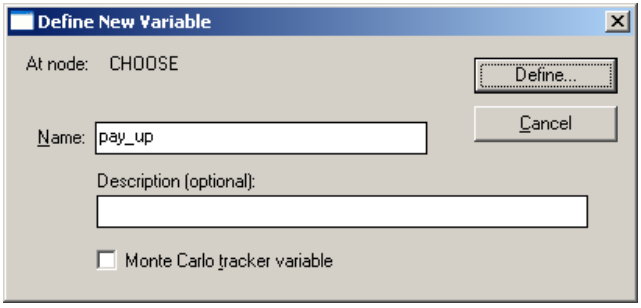
The previous section briefly introduced the use of the **Define Variable** commands in the right-click, quick menu. Now, try using these commands to define payoff variables.

Start by creating a variable called "pay_up" to represent the payoff of the Large increase node, and defining it at the root node (default for the tree) *before* using the variable in a payoff.

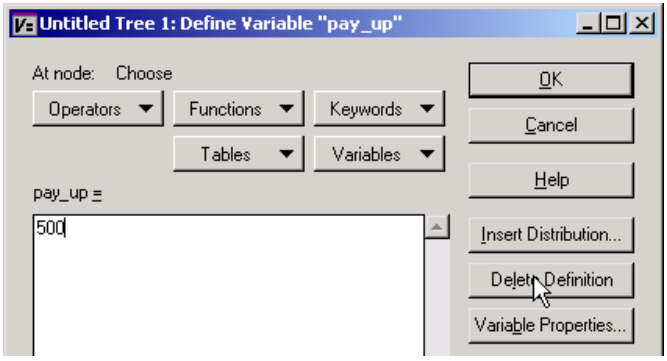
► **To define a new variable at a specific node: (*Method 2*)**

▷ Right-click on the root node, and choose **Define Variable >** New….

▷ In the dialog that appears, type `pay_up` for the new variable's name, and click **Define…** or press ENTER. A description for the new variable is optional.

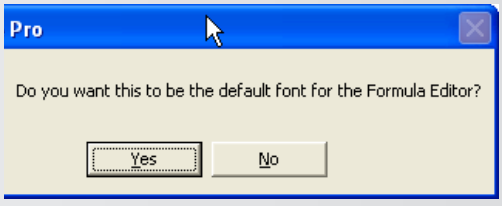


▷ In the Define Variable window that appears, enter the value of `500`, and click OK or press ENTER to return to the tree.



Tip:

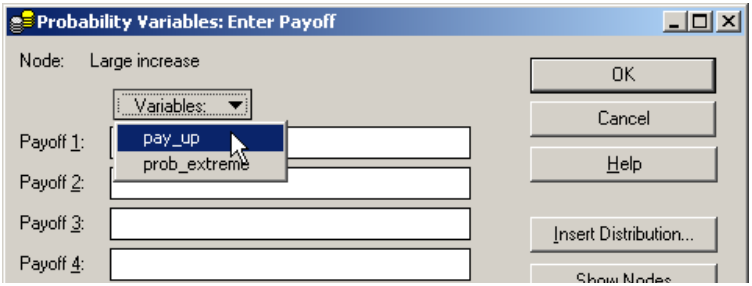
In v2007, the **Actions >** Font... command (also under the Display menu) can be used to change the default font used by the Define Variable editor. Choose an easier-to-read font size, for example.



If the display of variable definitions is turned on in the tree's preferences, this new variable's definition should be visible below the root node.

Now that the variable pay_up has been defined, use it in the appropriate tree calculation: the payoff at the Large increase node. Complete this step now.

- ▷ Either double-click on the Large increase node, or select the node and choose **Values >** Change Payoff….
- ▷ Delete the existing expression, the numeric 500, and either type `pay_up` or select the variable from the list labeled Variables. Click OK or press ENTER.



Now add a second payoff variable, for the Large decrease terminal node. Call the new variable pay_down — it will represent the payoff (loss) of this scenario.

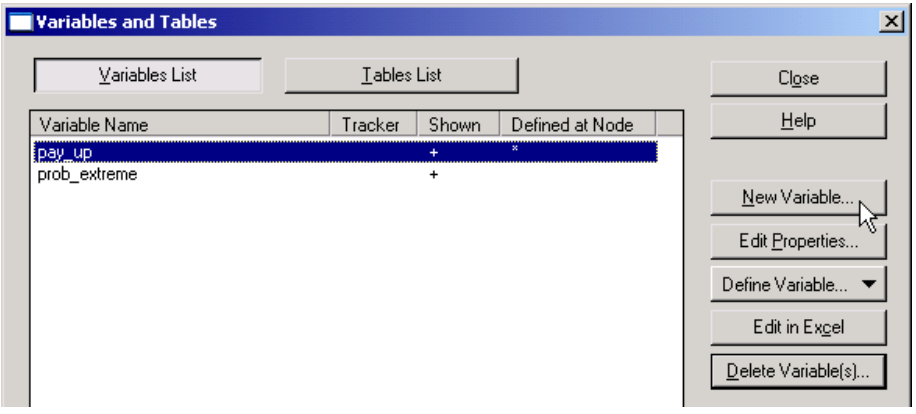
This time, use the Variables and Tables dialog to create the variable.

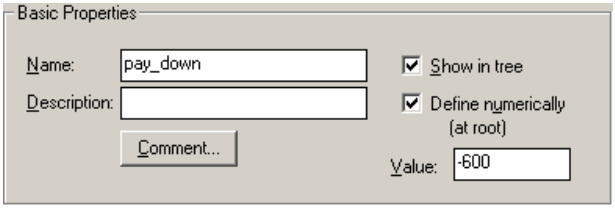
► **To define a new variable: (*Method 3*)**

- ▷ Choose **Values >** Variables and Tables..., or click on the **V=** button in the toolbar.



- ▷ Make sure the Variables List option is selected, and then Click the **New Variable...** button.





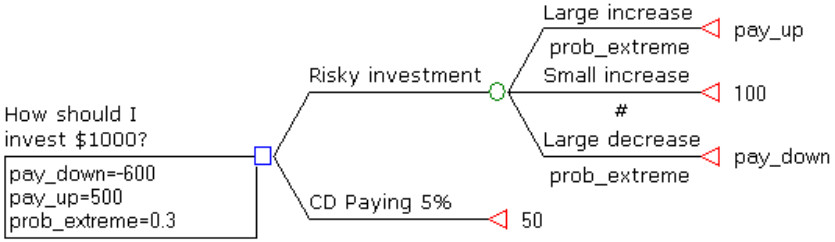
- In the Properties dialog for the new variable, enter the name `pay_down`. Check the option labeled Define numerically (at root) and enter `-600` in the Value box.
- Click OK or press ENTER to return to the Variables and Tables dialog.

The new variable, pay_down, should appear in the list of variables, along with prob_extreme and pay_up.

The payoff at the Large decrease node still needs to be updated, to use the variable pay_down.

- Click Close or press ESC to close to the Variables and Tables dialog and return to the tree.
- Either double-click on the Large decrease node, or select the node and choose **Values** > Change Payoff….
- Delete the existing numeric payoff, -600, and either type `pay_down` or select the variable from the drop-down menu labeled Variables. Click OK or press ENTER to return to the tree.

The updated tree is shown below.



Before continuing with the sensitivity analysis tutorial, take a moment to save the changes you have made to the Three Variables tree.



- Select **File** > Save.

Note:

Both the right-click menu and the Variables and Tables dialog can be used to define a variable at a node *other than the root node*. When first learning how to use variables, however, it is recommended that you make all numeric definitions at the root node. This will help avoid errors during sensitivity analysis, and when making changes to the definitions.

Performing one-way sensitivity analysis

TreeAge Pro can perform sensitivity analysis at a selected node using a range of values for a single variable (*one-way* sensitivity analysis) or across ranges of values for two or three variables simultaneously (*multi-way* sensitivity analysis). This chapter covers one-way sensitivity analysis. Multi-way sensitivity analysis, and other advanced sensitivity analysis topics, are described in Chapter 15.

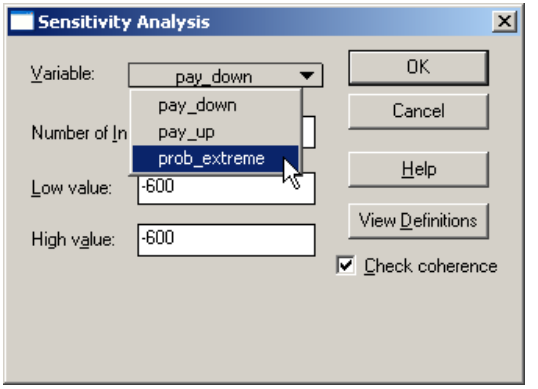
Start by analyzing the sensitivity of the decision to changes in the probability variable, prob_extreme.

► **To perform a one-way sensitivity analysis:**

▷ Open the Three Variables tree you created in the previous section (or find the copy of this file called Three Vars found in the Tutorial Examples subdirectory).

▷ Select the decision node.

▷ Choose **Analysis >** Sensitivity Analysis > One Way...., or click the toolbar button.



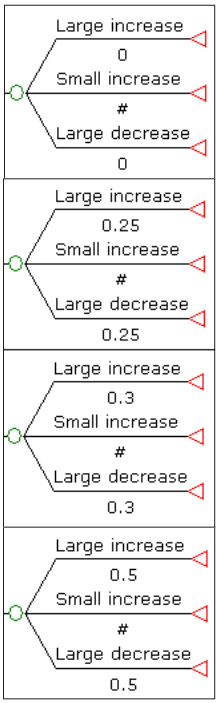
▷ In the Sensitivity Analysis dialog, click on the Variable pop-up menu, and change the variable to prob_extreme.

If you specified a low and high value in the variable's properties, that range will be shown. Otherwise, the baseline value will be shown.

Setting the sensitivity analysis range

The current definition of prob_extreme is 0.3. Recall from earlier in the chapter that our initial interest in sensitivity analysis with the Stock Tree is to test different probability assumptions for the risky investment. For instance, we would like to vary the probability estimates for the extreme outcomes at least from 0.25 to 0.3.

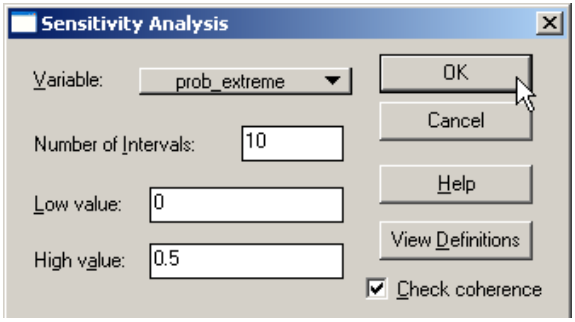
To make the sensitivity analysis more comprehensive, however, we could try the entire possible range for the variable prob_extreme. The tree fragments at left illustrate the concept.



Remember that prob_extreme is used in two branches, with the remainder (or complement) assigned to the middle branch. If we set prob_extreme to 0 (top subtree), the remainder calculated for the Small increase branch will be 1.0. So, prob_extreme=0 can be the minimum value for the sensitivity analysis.

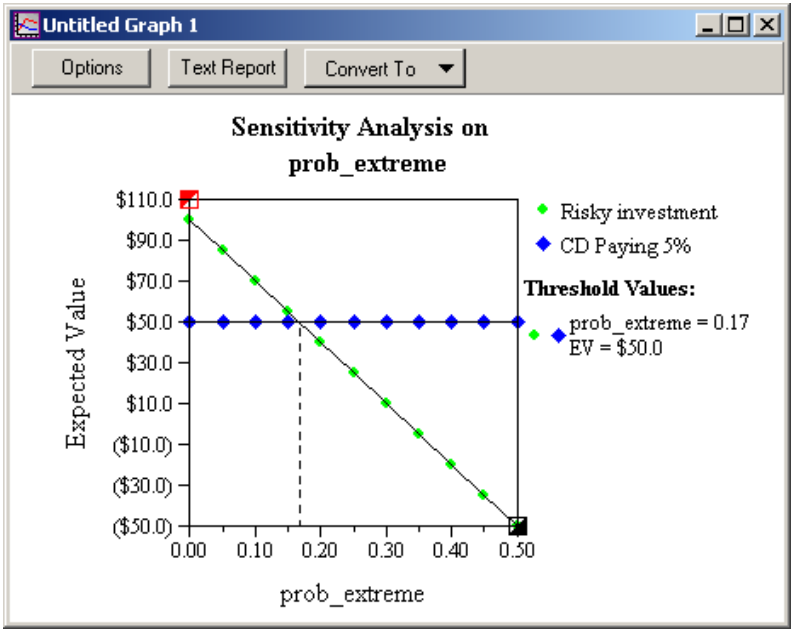
If we try to set prob_extreme to anything above 0.5, probability coherence errors will occur — for instance, at prob_extreme = 0.501 the extreme branches sum to greater than 1.0. The maximum possible value for the variable therefore is 0.5 (bottom subtree). The key estimates for prob_extreme, 0.25 and 0.3, are found within this range.

Set the sensitivity analysis range now.



- Type `0` for the Low value, and `0.5` for the High value.
- Change the number of intervals to `10`. Dividing the range into 10 intervals results in 11 recalculations at the decision node, for prob_extreme equal to 0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, and 0.5.
- Press ENTER or click OK to run the analysis.

TreeAge Pro should immediately begin the analysis. If the analysis were a long one, you could monitor its progress in the status bar at the bottom of the TreeAge Pro window. When it is complete, a graph is displayed.



If the analysis does not complete due to errors, read the error message and make note of the node where TreeAge Pro reports a problem. Most likely, the problem is either to wide a range for the uncertain variable, or a problem with the assignment of probability variables in the chance node branches.

The sensitivity analysis graph and report

Because the analysis was done at the decision node, there should be two lines corresponding to the two alternatives, Risky investment and CD paying 5%. Each alternative's expected value is plotted as a function of the increasing value of prob_extreme.

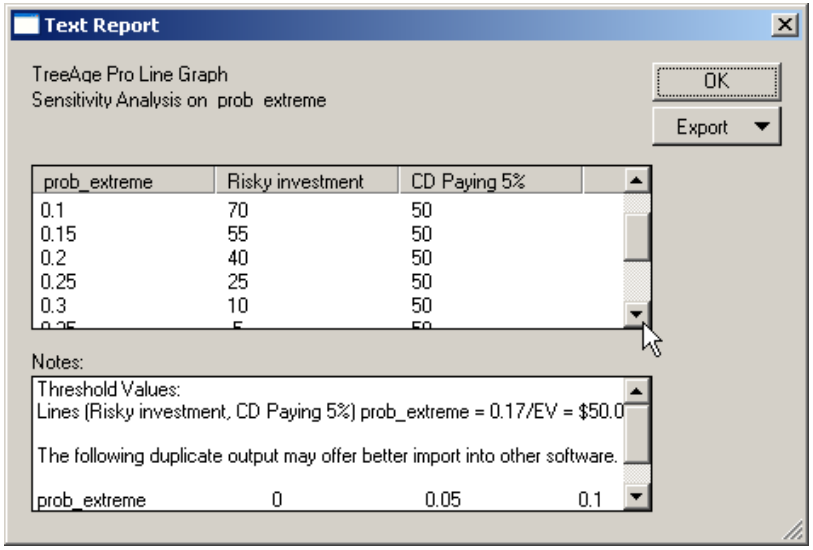
Each strategy's line is composed of line segments connecting the line marker symbols that identify that alternative's expected value at successive intervals of the analysis. A legend to the right identifies the symbol assigned to each particular alternative, and any threshold values (see next page).

Deviations of a line from the horizontal indicate that strategy's sensitivity to the variable. An alternative represented by a horizontal line in the graph, such as the CD paying 5% option in the example, is unaffected by the changes in the variable. In contrast, the payoff represented by the Risky investment line is an decreasing function of the variable.

The analysis text report, showing the underlying calculated values, can be opened by choosing **Actions >** (or **Graph >**) Text Report….

Graph tips:

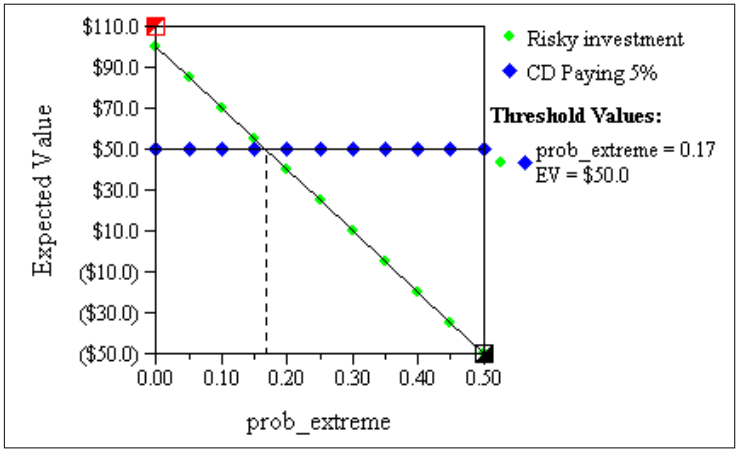
- If the trend of a line in the graph does not make intuitive sense, this may indicate a problem with the definition or use of the variable in the model.
- Sensitivity analysis can be used to look also for errors in complex formulas, which might be indicated if changing the value of a parameter (even one that is certain) does not have the anticipated effect on calculations.
- The visual elements of the one-way sensitivity analysis graph, like other line graphs, can be customized in a number of ways, as described in Chapter 6. This includes changing graph size, texts, line markers, and numeric formatting. The **Options** button and **Actions** and **Graph** menus provide access to many customization options.



Sensitivity analysis thresholds

The sensitivity analysis results can be interpreted graphically.

If two lines in the graph intersect, at the corresponding value of the variable these two alternatives have the same expected value. Crossing points that represent a change in the optimal strategy are called *thresholds*. From the standpoint of expected value, the decision maker should be indifferent between the two options at a variable's threshold value.



The Threshold Values legend

The sensitivity analysis graph will, if necessary, include a Threshold Values legend which identifies thresholds, displaying:

- the symbols for the lines which cross
- the value of the variable at the crossing point
- the expected value at the crossing point

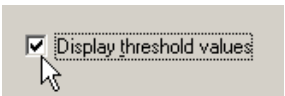
In the graph, at every threshold point a dashed line drops down to the horizontal axis.

Interpretation usually requires simply looking at what happens to the left and right of the threshold. In the example analysis on prob_extreme, the threshold legend identifies a crossing point when prob_extreme = 0.17. For values higher than that, including the baseline 0.3 probability, the CD has a higher return and is therefore optimal. For values of prob_extreme less than 0.17, Risky investment is optimal.

In this analysis, the threshold is not close to our two best estimates, 0.25 and 0.3, so perhaps the model is not sensitive to this particular uncertainty.

The only aspects of the threshold legend which can be modified (indirectly) are the two numeric formats; if you change the numeric format of either axis, the format of the corresponding item in the threshold legend will be adjusted automatically. None of the threshold legend text may be directly changed, as these are calculated values.

Threshold information may also be hidden by opening the Options dialog, and unchecking the Display threshold values setting. This will hide both the dotted threshold lines and the threshold legend.



A caveat on thresholds

If all of the lines in a sensitivity analysis graph are straight, the threshold analysis in the graph will be exact. However, if any lines appear curved, the threshold analysis is a linear approximation, and its accuracy will increase as the width of the intervals decreases.

The analysis performs calculations only at the ends of the *N* number of intervals specified when running the analysis. The lines plotted on the graph are accurate at these *N+1* discrete points, but not necessarily in between (e.g., if an alternative's expected value is an exponential or other non-linear function of the variable's value).

The sensitivity analysis graph will *not* recognize a threshold if the optimal strategy is the same at both ends of the interval, but changes back and forth within the interval. The likelihood of this error can be reduced by increasing the number of intervals, thus reducing their width.

> ***Tip:***
> A more sophisticated, non-graphical form of threshold analysis is described in Chapter 15. This chapter also includes more information on one one-way sensitivity analysis and other kinds of sensitivity analysis.

Chapter 13

Working with Variables

This chapter expands on the aspects of TreeAge Pro's variables interface introduced in Chapter 12, and also introduces a number of important windows, dialogs, and other tools that facilitate working with variables in trees.

For information on managing tables using the Variables and Tables dialog, see Chapter 18. Also see Chapter 14 on building complex variable definitions and using functions. Users of the **Excel/COM module** can refer to Chapters 27, 28, and 30 to learn about special module features available for managing variables.

In this chapter:

- Variable Categories 154
- Variables Report 157
- Variables testing tools 158
- Formula Editor and Insert Variable dialog 160
- Variable Definitions window 161
- Variables and Tables list 164
- Variable Properties 166
- Define Variable window 168

Variable Categories

To help organize long lists of variable definitions in a tree, use the Variable Categories dialog to create a hierarchical structure of variables. The categories can be used both in the display of variable definitions in the tree window (under nodes/branches), as well as in the properties grid.

The tree's variables display preferences allow you to turn on and off the categorization of variables.

The Variable Categories dialog can be opened from either the **Values** menu, or from the Variables and Tables dialog.

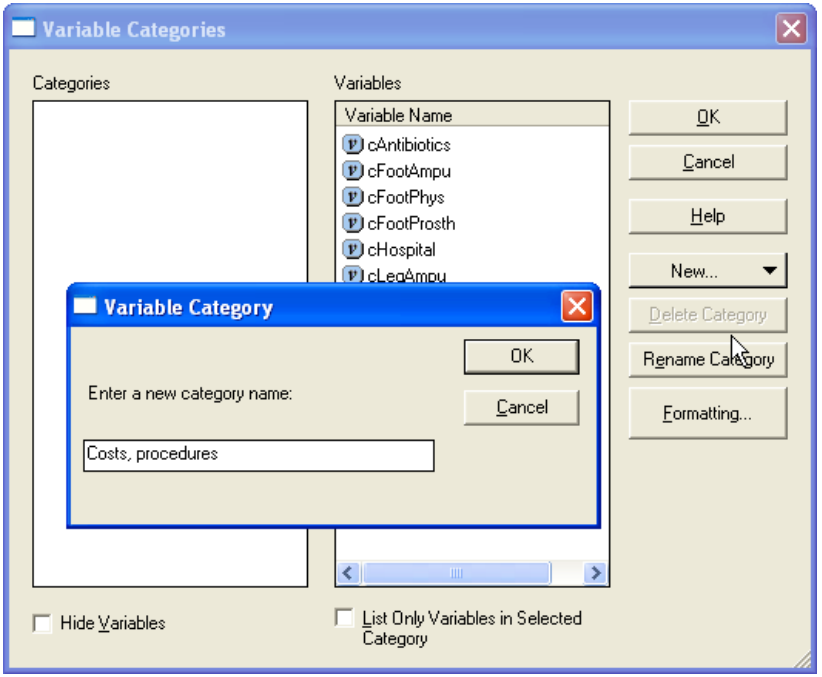
► **To create a list of variable categories and sub-categories:**

▷ Choose **Values** > Variable Categories….

▷ Click on the button labeled **New Top-Level Category**. Fill in a name for the new category and click OK.

▷ To rename an existing category, select it and then click the **Rename Category** button.

▷ To create a sub-category, select an existing category and then click the button labeled **New Sub-Category**.



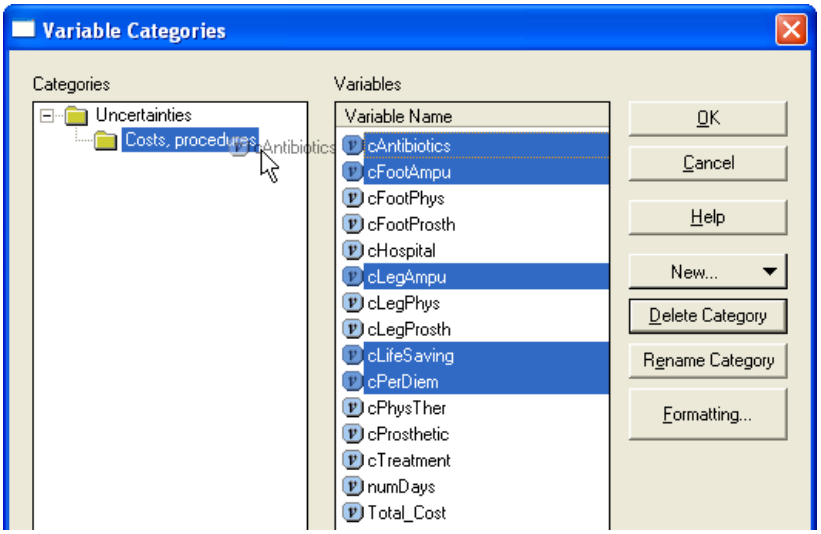
▷ To get rid of a category (including all of its contained sub-categories), select it and click the **Delete Category** button.

▷ To filter the lists of variables, use the checkbox options below each list.

► **To add variables to categories:**

▷ To add a single variable to a category, select it in the list on the right and drag it to a category or sub-category on the left.

▷ To select multiple variables to add to a category, either click and drag to create a selection rectangle touching a set of variables, or shift-click on variables to add them to a selection.



▷ To move a variable or sub-category from one category to another, simply drag it.

What is the appropriate treatment for this patient?

[+]

```
--- Uncertainties
   --- Costs, procedures
       cAntibiotics=500
       cFootAmpu=5,000
       cLegAmpu=10,000
       cLifeSaving=10,000
       cPerDiem=1,000
cFootPhys=10,000
cFootProsth=2,000
cHospital=cPerDiem * numDays
cLegPhys=25,000
cLegProsth=10,000
cPhysTher=0.
cProsthetic=0.
cTreatment=0
Total_Cost=cHospital+cTreatment+cProsthetic+cPhysTher
```

Variable categories notes:

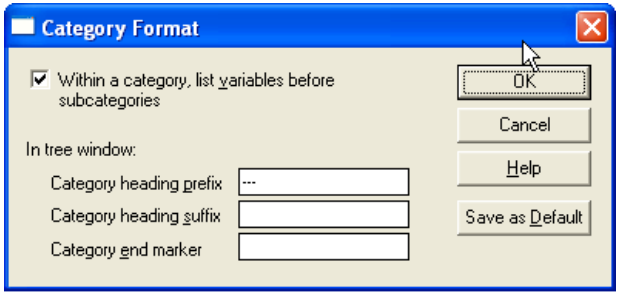
- Categories (and variables/subcategories within them) are sorted alphabetically
- Uncategorized variables can appear before or after categories (see Formatting, next page)
- Category prefix and suffix can be changed (see next page)
- The categorized display of variables can be turned on/off in the Preferences dialog (without deleting categories).

Variable categories preferences

Formatting options can be used to change how variables categories are displayed in the tree, under nodes/branches.

► **To change category formatting:**

▷ Go to **Values >** Variable Categories…, and click on the **Formatting...** button.



The checkbox in the Category Format dialog allows you to control, within a category, whether variables are listed above or below subcategories. (Variables that are not included in any category are always sorted below any categories.)

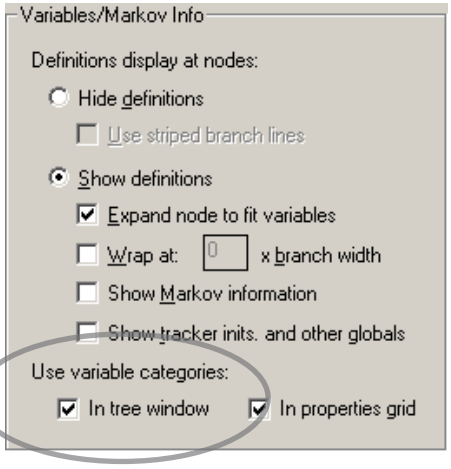
The other settings in the Category Format dialog can be used to add/remove prefix or suffix characters to all variable category headings in the tree display.

If the display of variable definitions is turned on in tree, you can turn on and off the categories display in the Preferences dialog (without deleting your categories).

► **To disable variable categories in the tree display:**

▷ Select **Edit >** Preferences…, and choose the Variables Display category from the list.

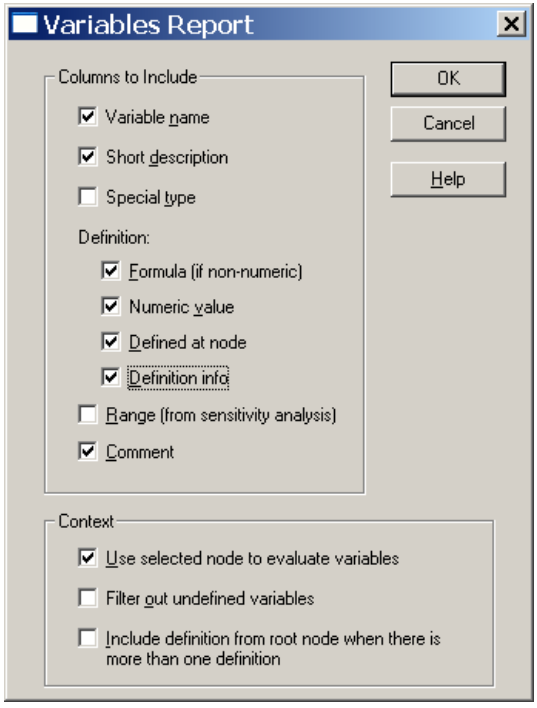
▷ Under Use variable categories, uncheck In tree window.



Variables Report

The Variables Report can be used to display and export two types of variable lists:

1. A complete list for the the tree including each variable's name and selected properties (description, comment, sensitivity analysis range, numeric values, and formula).
2. For a single selected node, a list of variables defined at or to the left of the node and their properties. (New in v2007)



► **To create a variables report:**

▷ Choose **Values >** Reports > Variables….

▷ In the Variables Report setup dialog, specify the information to include, then click OK or press ENTER to display the report.

The report is displayed initially in a text report dialog, and can be copied (or exported) into a spreadsheet or other document.

Variables reporting options and calculations

By changing the report options, you can control what is included in the report. For example:

- Unchecking the Variable name option will result in variables being identified by their short description in the report.
- To show only numeric values in the report, select the option labeled Numeric value, and uncheck the option labeled Formula.

Changing the options under Context can affect which definitions appear in the report:

- Use selected node... and Include definition from root... are mutually exclusive.
- Filter out... will remove from the report variables with formulas that cannot be calculated (in the chosen context).
- If Filter out... and Use selected node... are checked, the report includes only variables that can be used at the active/selected node (and its subtree).
- If Include definition from root... is checked, the context for the calculation of a formula definition is the location of the definition. (This is unlike a real tree analysis, in which case a payoff/reward node or chance node branch is usually the context.)

Variables testing tools

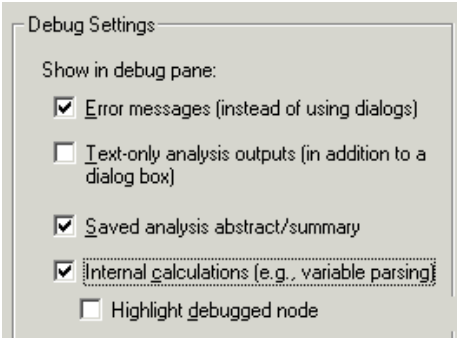
In addition to using TreeAge Pro's Analysis menu commands to see the results of node calculations, TreeAge Pro also provides tools for testing individual variables.

Debugging variable calculations during analysis

The Debug text pane can be used to report on every variable calculation during an analysis. This feature can be used to search for problems in complex formulas.

► **To turn on detailed calculation debugging:**

▷ In the tree, choose Edit . Preferences…, and select the Debugging category.


Debug Settings
Show in debug pane:
Error messages (instead of using dialogs)
Text-only analysis outputs (in addition to a dialog box)
Saved analysis abstract/summary
Internal calculations (e.g., variable parsing)
Highlight debugged node

▷ Select the option labeled Internal calculations….

Reporting all variable calculations will slow down calculations, so this setting should be turned off except when debugging.

See Chapter 21 for more tips on using the Debug pane preferences.

The Calculator/Evaluator

The Calculator/Evaluator is designed for testing how variables and formulas used in a tree will calculate at a specific node. It will perform the calculation as if it were a payoff or probability calculation occurring at the selected node during roll back. Distributions will return their mean values, and Markov counters their default values.

► **To calculate a variable or formula at the selected node:**

▷ Edit a probability, payoff, or other expression.

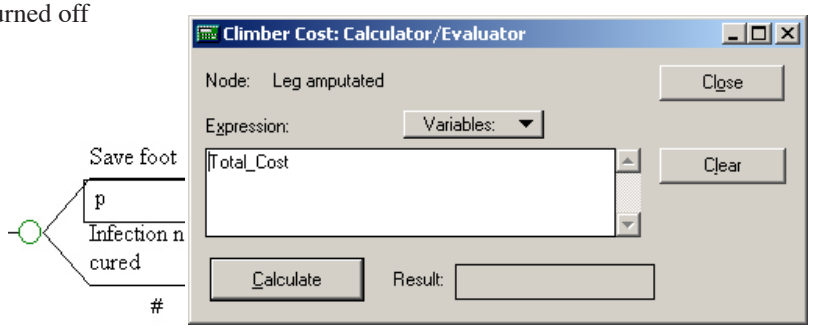
▷ Choose **Values >** Open Calculator/Evaluator…, or click on the "equals" tool bar button..

▷ If activated from a probability, payoff, or other expression editor in the tree, that formula will appear in the Expression box. You can alternatively pick a variable name or enter a different formula.

▷ Click on the Calculate button.

If the expression can be calculated at the selected node, the result will be displayed. Errors that occur during calculations will be reported.

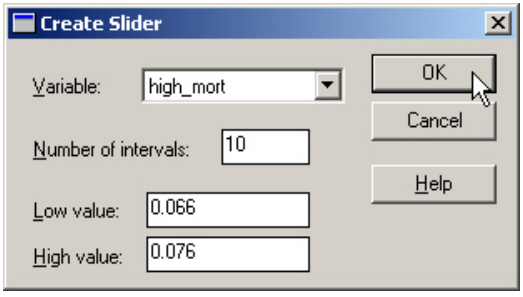
Changing the selection in the tree window while leaving the Calculator/Evaluator open allows you to quickly test the same calculation at different nodes.

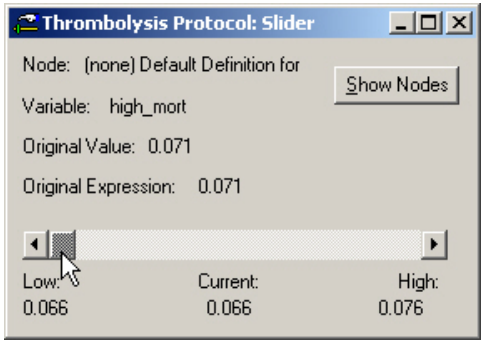


Variable sliders

As described in Chapter 12, sensitivity analysis automatically recalculates a model for a range of values of a variable. TreeAge Pro also makes it easy to *manually* increment a variable's value over a specified range using a tool called a Slider.

A slider can be used to quickly modify one or more variable's values, without having to repeatedly open Define Variable windows. At each interval, you can perform a specialized calculation or analysis that is not part of a regular sensitivity analysis — for example, using the Calculator/Evaluator or generating a probability distribution.





► **To create a slider to control a variable:**

▷ Select the node at which a variable is defined and choose **Values >** Create Slider….

▷ In the Create Slider dialog, select a variable from the pop-up menu and specify a value range and a number of intervals. Click OK to display the Slider tool.

▷ To use the Slider, click a directional arrow or drag the sliding bar to change the value of the variable within the specified range.

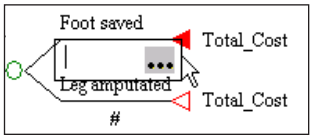
With the Slider open, you can perform any analysis on the model (including roll back) based on the value of the variable shown in the Slider. The changes to the variable's value will apply only until the Slider is closed or a regular sensitivity analysis is performed on the chosen variable.

If you click the Show Nodes button, TreeAge Pro will activate the tree window and reselect the node on which the slider is operating.

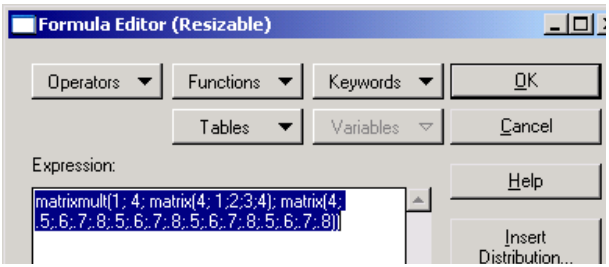
> ***Tip:***
> For another kind of manual sensitivity analysis, see the information on stored analysis sequences in Chapter 19.

Formula Editor and Insert Variable dialogs

When the text cursor is in a probability editor, an ellipsis (…) button appears. This button opens the Formula Editor.



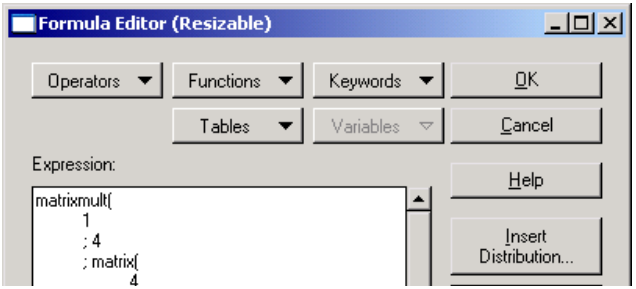
The resizable Formula Editor dialog includes formula-building popup menus designed to make constructing complex formulas easier.



The dialog can also be opened from edit fields other than probabilities using the "…" button on the tool bar, or the **Values** menu command).



If you have a very complex expression, with nested functions, use the **Actions >** Indent Functions command to display the expression in an outline-type format.

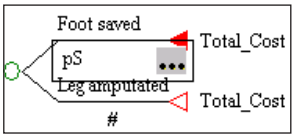


Refer also to the section on the Define Variable window, later in this chapter for more tips on the Formula Editor.

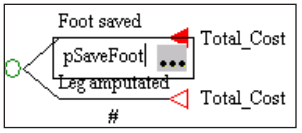
Inserting variable names

Alternatively, the Insert Variable dialog can be used to help enter the name of an existing variable into a formula.

If you know the first few characters of the variable (or table) name you wish to insert, you can optionally type them prior to choosing **Values >** Insert Variable (or using the keyboard shortcut CTRL+i).



If only one variable name matches what you have typed so far, that variable name will be inserted automatically (without displaying the Insert Variable dialog).



Variable Definitions window

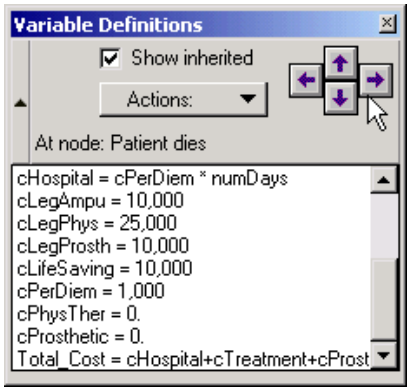
The Variable Definitions window is a powerful tool that includes variables-related functionality not found in TreeAge Pro's other variable tools. It is linked to the variable definition boxes displayed below branches in the tree, also providing a view of the definitions at a node. In addition to displaying definitions at the selected node, the Variable Definitions window can be used to:

- edit an existing definition by double-clicking on it
- define a new or existing variable using the **Actions** popup menu
- cut, copy, or paste selected definitions using the **Actions** menu
- show "inherited" definitions in the path back to the root node
- view tracker variables' initial values
- view and edit named distributions

There are a number of ways to open the Variable Definitions window at a node.

► **To open the Variable Definitions window:**

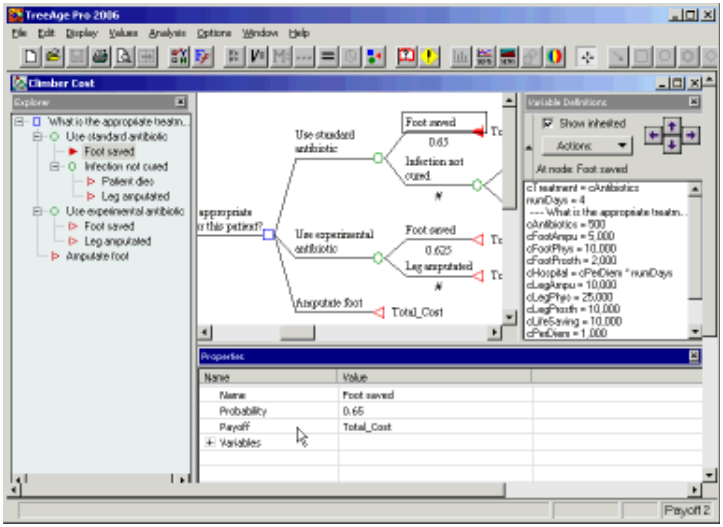
▷ If variable definitions are displayed below a branch in the tree, double-click on the definition box to open the Variable Definitions window.



▷ The Variable Definitions window can also be opened by choosing **Values** > Show Variable Definitions Window (or clicking on the double *V*= tool bar button).



The Variable Definitions window can be resized and docked, like the explorer pane and properties grid. Its position and size are stored with the tree.



Updating and adding variable definitions

Using the Variable Definitions window, you can edit, add, or remove single definitions at the selected node, as well as move to a different node.

► **To edit an existing variable definition:**

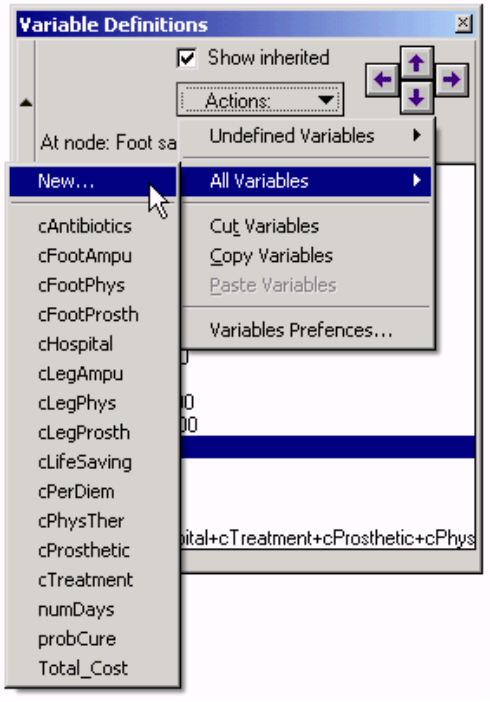
▷ Scroll down if necessary, and double-click on a definition in the list to open a Define Variable window (see below).

The **Actions** menu includes separate sub-menus listing named variables not yet defined at the node, and all named variables.

► **To define an existing variable name at the node:**

▷ Select the variable name from either the Undefined Variables list or the All Variables list..

The All Variables sub-menu includes a New… command, which can be used to add a new variable name to the tree.



Changing the node selection

The navigation arrow buttons at the top of the Variable Definitions window work like the arrow keys on the keyboard. The buttons allow you to move the selection one node at a time in any direction, in order to view/edit that node's list of definitions.

Using the Show inherited option

The Variable Definitions window always lists the definitions at the selected node. It can also show definitions made at nodes to the left, and global definitions of tracker variables and distributions.

► **To display inherited definitions and global values:**

▷ Check the Show inherited option at the top of the Variable Definitions window.

If variable definitions exist at a node to the left, the definitions from that node are appended to the list, prefixed by a dashed line and the name of the node. The window will continue listing nodes and their definitions in the path back to the root node.

The window will show only the first definition of a variable found in the path back to the root node.

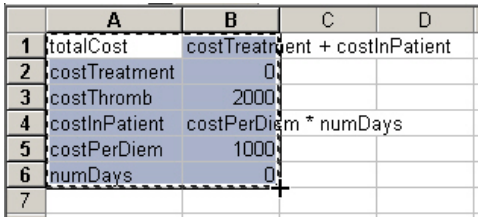
```
cPhysTher = cLegPhys
cProsthetic = cLegProsth
cTreatment = cAntibiotics + cLegAmpu
numDays = 10
 ----- What is the appropriate treatm...
cAntibiotics = 500
cFootAmpu = 5,000
cFootPhys = 10,000
cFootProsth = 2,000
cHospital = cPerDiem * numDays
cLegAmpu = 10,000
cLegPhys = 25,000
cLegProsth = 10,000
cLifeSaving = 10,000
cPerDiem = 1,000
Total_Cost = cHospital+cTreatment+cProsthetic+cPhysTher
```

At the bottom of list will be displayed any globally-defined names — either tracker variables, or distributions having a variable-type name.

Cut, copy, and paste variable definitions

Groups of variable definitions (or single definitios) can be cut or copied from node to node, tree to tree, or between a text editor and a tree, using the Variable Definitions window.

New variables can be added to a tree by pasting into the Variable Definitions window, and existing definitions at a node can be updated.



	A	B	C	D
1	totalCost	costTreatment + costInPatient		
2	costTreatment	0		
3	costThromb	2000		
4	costInPatient	costPerDiem * numDays		
5	costPerDiem	1000		
6	numDays	0		
7				

Definitions are placed onto the clipboard as tab-delimited text, with a column for variable names and another for definitions (with no "=" between them).

► **To copy or cut variable definitions from a node:**

▷ Open the Variable Definitions Window.

▷ To copy or cut a subset of the definitions shown, hold down the CTRL key while selecting individual definitions.

▷ Using the **Actions** menu, select Copy Variables or Cut Variables.

▷ If you do not select any items in the list, the Copy Variables command will include all definitions, while the Cut Variables command will be unavailable.

▷ If the Cut Variables command is used, the selected definitions are placed on the clipboard and *removed* from the node.

If you paste into a spreadsheet, you can make changes to definitions and then copy the two columns (names and definitions) to be pasted back into the Variable Definitions window.

The Paste Variables command is available when the Variable Definitions window is open at a node and a well-formed, tab-delimited list of definitions has been copied onto the Windows clipboard.

► **To paste variable definitions at a node:**

▷ Open the Variable Definitions window at a node, and choose **Edit >** Paste Variables.

Definitions pasted from the clipboard will overwrite any existing definitions of the same variables found at the selected node.

If a variable in the pasted list does not yet exist in the tree, it will be created.

Variables and Tables dialog

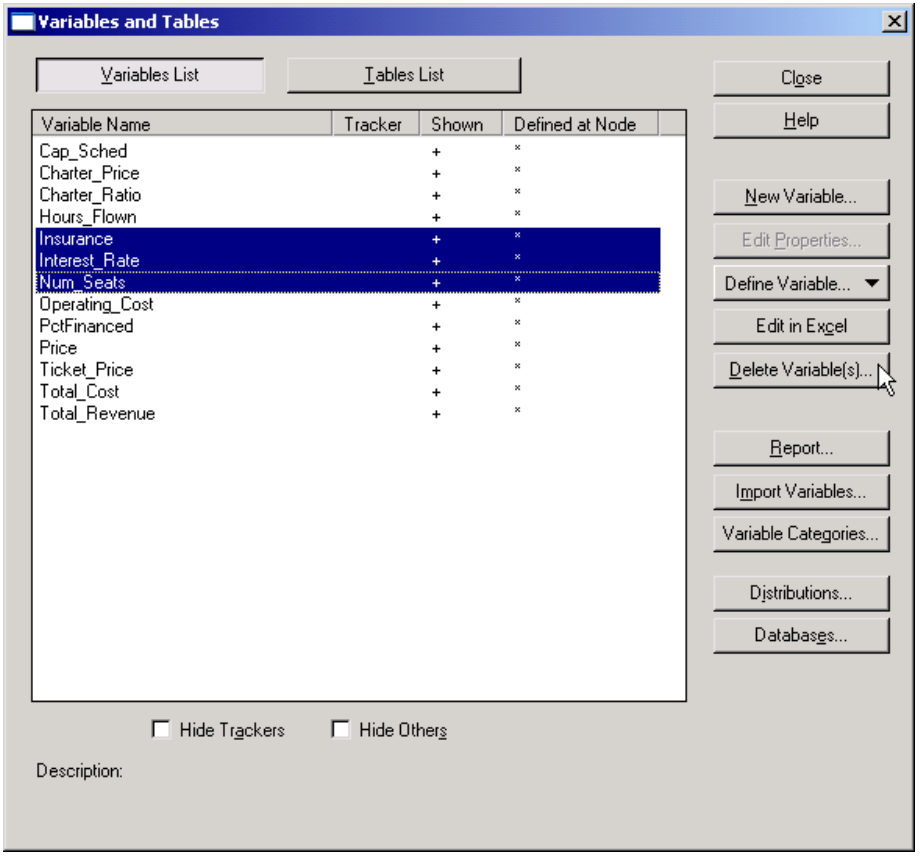
Numerous methods exist for creating and defining variables in TreeAge Pro, for example:

- type a new variable name directly in a probability or other expression
- right-click on a node and choose Define Variable...
- use the Actions menu in the Variable Definitions window (see previous page)

This section details the use of the Variables and Tables dialog for *variables* management tasks, including:

- replacing the list of variables in a tree with a list imported from a different tree
- deleting a variable name from the tree
- renaming a variable (throughout the tree)
- defining multiple variables at once
- defining a variable at multiple selected nodes

This section does not cover *tables* management tasks. See Chapter 17 for complete instructions on working with the Tables List.



► **To modify or remove a definition of a variable:**

▷ Select the node at which the variable is currently defined, and choose **Values >** Variables and Tables.



▷ In the Variables List, select the variable (or variables), click **Define Variable…**, and choose either At Selected Node or Default for Tree.

▷ Make changes in the Define Variable window and press ENTER or click OK to close the window.

▷ To remove the definition from the node, click the Delete Definition button.

Deleting a definition of a variable at a node does not delete the variable name from the tree.

It is possible for a variable name that you have created to be deleted from the tree completely, along with any numeric/formula definitions of the variable.

► **To delete a variable and all of its definitions from a tree:**

▷ Select the variable name in the Variables List and click the **Delete…** button.

Deleting a variable removes all *definitions* of the variable from the tree. *References to the deleted variable will not be removed* from payoffs, probabilities, and other variable definitions. If such formulas are not corrected, your model may generate an error message upon calculation.

See Chapter 12 for information on using the **Options >** Find/Replace dialog to search for any references to a name in formulas throughout the tree.

Selecting multiple variables

There are two ways to select multiple variables in the Variables List (e.g., in order to define or delete a set of variables).

► **To select any subset of variables:**

▷ Hold the CTRL key while clicking, one after another, on each of the variables to be selected.

► **To select a contiguous set of variables:**

▷ Click on the first variable in a group and hold down the SHIFT key while clicking on the last variable.

Importing a list of variables from another tree

Clicking the **Import…** button in the Variables List will prompt you to delete the current list of variable names in the tree, and replace it with a list of variables from another saved tree. Unlike using the Delete… button, however, no variable definitions will be removed (including default numeric definitions at the root node).

See the section on the Variable Definitions window earlier in this chapter for information on copying and pasting one or more variable *definitions* between trees.

Edit in Excel (Excel module)

If your license includes the Excel module, the **Edit in Excel** button in the Variables and Tables dialog can be used to export selected variables to a table in a new Excel worksheet, where they can be edited.

From Excel, the add-in's **TreeAge >** Add or Update Variables menu command can be used to update the active tree with the updated list of variables. See Chapter 29 for details.

Variable Properties dialog

The Variable Properties dialog appears when you create a new variable either using the Variables and Tables dialog, or by typing a new variable name directly in a probability or other expression. You will also see this dialog when you select an existing variable and click the **Edit Properties…** button in the Variables and Tables dialog.

One important use of the Properties dialog is for renaming a variable (as an alternative to deleting a variable from the tree and creating a new one with the desired name).

► **To rename an existing variable:**

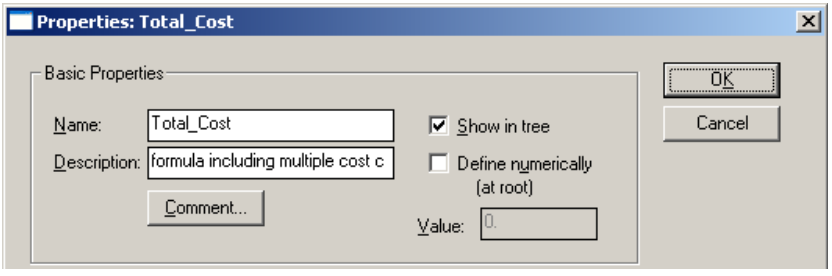
▷ Select the existing variable in the Variables and Tables list and click the Edit **Properties…** button.

▷ Change the Name property, and close the dialog.

The name change will cascade throughout the tree, updating all definitions of the variable, *as well as all references to the variable*. This means that you will not need to do anything else to update the tree.

Variable properties are divided into three groups: Basic Properties, Properties for Sensitivity Analysis, and Tracking Properties (for microsimulation).

Each variable has three basic text properties: the variable name, a short description, and a long comment.

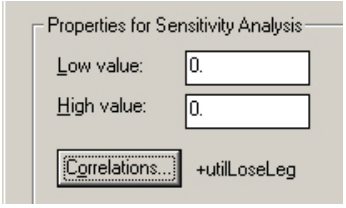


The variable's name must conform to naming rules, as described in Chapter 13. The short description is optional; it is used in graphs. The long comment can be used to hold notes that memorialize a variable's meaning or sources.

The Basic Properties group also includes a check box labeled Show in Tree, which is checked by default. If you clear the check box, definitions of this variable will remain hidden when other variable definitions are displayed in the tree.

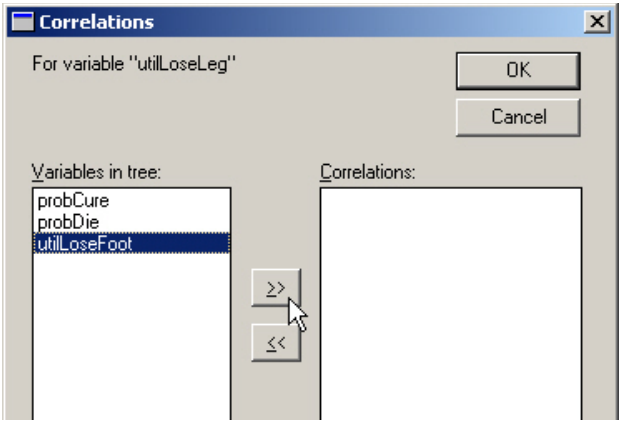
If the option labeled Define numerically is checked, the number entered in the Value editor will be used to define the variable at the root node.

Low and high values entered in the sensitivity analysis section of the dialog will be used as the suggested range for the variable in sensitivity analysis setup dialogs.



Sensitivity analysis correlations

Correlations can be set up between any number of pairs of existing variables. When a sensitivity analysis is performed on a variable correlated to another variable, the option is presented to simultaneously vary correlated variables over their own value ranges.



► **To define variable correlations:**

▷ Click on the Correlations... button in the Properties dialog.

▷ In the Correlations dialog, select the variable to correlate with from the list on the left, and click the >> button to move it into the list on the right.

▷ In the New Correlation dialog, select the appropriate type of correlation (either perfectly positive or negative). Click OK to return to the Correlations dialog, and then close it.

The linked variables will now be listed to the right of the Correlations button, with plus or minus symbols indicating the type of correlation. The identical correlation will show in the properties of the two correlated variables. The correlation can be modified or removed from either variable's Properties dialog.

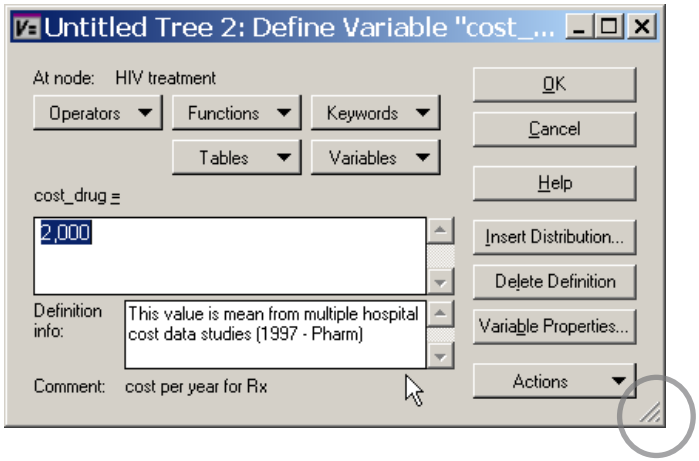
► **To remove a correlation:**

▷ In the Properties dialog for either of the correlated variables, click on the **Correlations...** button.

▷ In the Correlations dialog, select the correlated variable from the list on the right, and click the << button to remove it from the list.

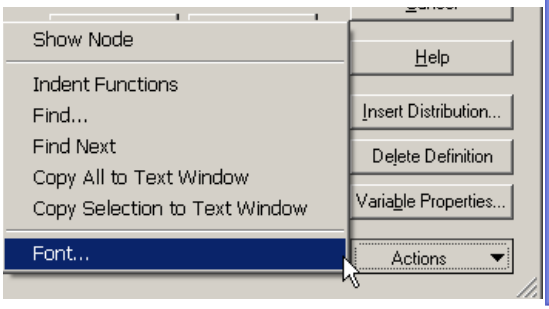
To change a correlation's type (e.g., from negative to positive), you must remove the existing correlation and recreate it with the proper correlation.

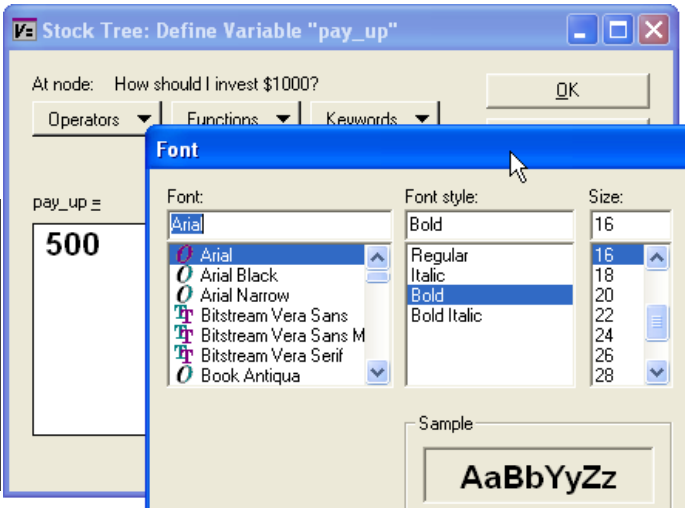
The Define Variable window



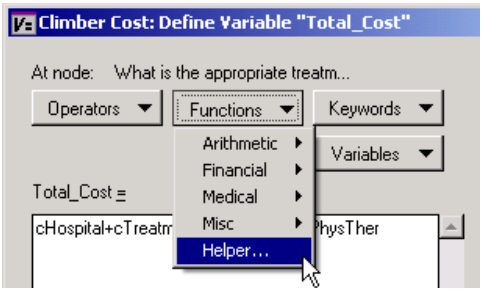
The Define Variable window provides a variety of aids for building complex definitions (like the Formula Editor, described earlier), including:

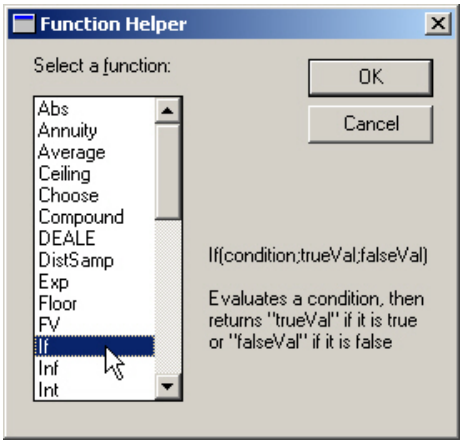
- TreeAge Pro 2007 adds the ability to annotate complex definitions. The optional Definition Info is associated with the definition at the node (as distinguished from a general description/comment assigned to the variable in the Properties dialog.)
- The window is resizable. If you are working with very long definitions, either maximize the window or use the resize control in the bottom right corner.
- Use the Indent Functions command in the **Actions** menu to automatically break complex, nested functions onto multiple lines for easier reading. (Tabs and carriage returns will be removed when the window is closed.)
- The **Actions** menu includes a Font... command can be used to change the default font used for the Define Variable editing window. Choose an easier-to-read font size, for example.





See Chapters 14 and 21 for more information on working with complex variable definitions.





Using the Functions Helper

The Define Variable window's Functions pop-up menu lists TreeAge Pro's many built-in functions, and includes a Function Helper to walk you through the steps of properly setting up any function.

TreeAge Pro includes a variety of mathematical, financial, logical, and other functions, like If(), NPV() and Discount().

To use a function in a formula, simply type its name followed by the required parameters in parentheses (like in Excel). For functions which allow multiple arguments, the arguments must be separated by semicolons (";").

Like variables, functions' names are *not* case-sensitive and, in most cases, take arguments inside the parentheses.

See Chapter 14 for full details on all available functions in TreeAge Pro.

The Define Variable window's Functions pop-up menu lists TreeAge Pro's many built-in functions, and includes a Function Helper to walk you through the steps of properly setting up any function.

You select an appropriate function, and TreeAge Pro will prompt for the requisite list of arguments, and enter the correct formula for you.

User-defined Python functions (new in v2007)

Starting with TreeAge Pro 2007, user-defined functions can be created within variable definitions, using the Python script language.

User-defined functions are entered as variable definitions, using a regular variable and the Define Variable window. Python functions require indenting and carriage returns, so the Define Variable window's behavior changes slightly when it recognizes you are entering/editing a user-defined function.

See Chapter 21 for more details.

Chapter 14

Building Formulas Using Variables and Functions

This chapter focuses on how to use variables to build complex formulas, in particular for payoff calculations.

Chapter 12 covers using variables for the purposes of sensitivity analysis on parameter uncertainties in the model.

In this chapter:

- Variable formulas 172
- Using mathematical, statistical, and other functions 176
- Arithmetic functions 177
- Financial/discounting functions 178
- Miscellaneous functions 179
- Using recursive variable definitions 180

Variable formulas

Chapter 12 explained some of the advantages of using variables, rather than numeric point values, in decision trees. Chapter 13 included details on the tools (e.g., windows, dialogs) available in TreeAge Pro when working with variables in decision trees.

This chapter will continue the tutorial on variables by explaining, in detail, the logic that dictates where variables should be defined in a tree and how variables are evaluated during tree calculations. Provided a basic understanding of how TreeAge Pro searches for variable definitions, it will become easier for you to identify the best location for each variable definition you create.

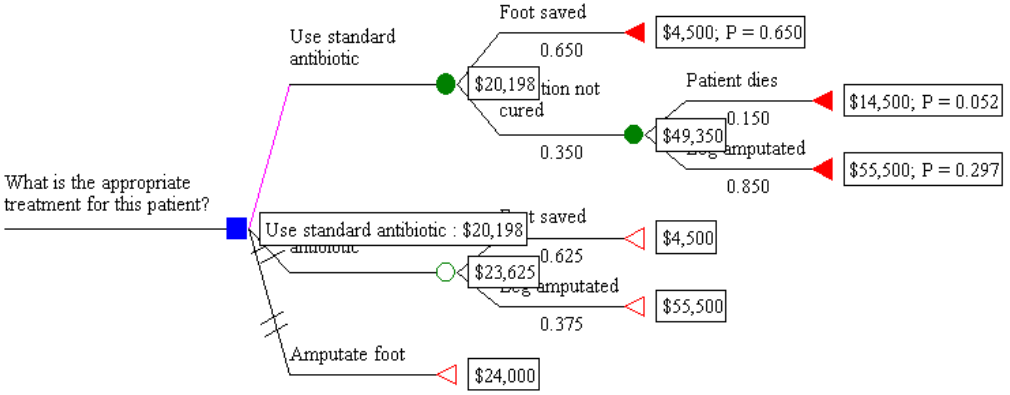
For information on working with variables in influence diagrams, see Chapters 25 and 26.

Building a complex cost formula: an example

The example model used in this chapter is based on the following medical treatment scenario. The assumption is that you are evaluating a new pharmaceutical treatment used in your hospital for advanced infections in an diabetic population. Initially, the modeling of the new intervention is being done on a cost basis, looking only at foot infections.

Previously, the treatment decision was between prompt amputation of the foot, and a course of high-dose, intravenous antibiotics. The new intervention is an antibiotic that works much faster that the old drug, so you will know sooner if it is going to halt the infection and save the foot; soon enough to have complete certainty that performing an amputation below the knee will avoid any mortality risk. For this reason, the new drug has already been adopted.

However, the new drug actually halts fewer infections over its shorter course of treatment, which results in a higher number of amputations. Based on the experience of your medical center, you estimate the costs and probabilities and put them into a decision tree, shown below.



A copy of this tree file, called Cost Formula, is found in TreeAge Pro's Tutorial Examples subdirectory.

As shown in the model, the original high-dose, intravenous antibiotics combined with surgical debridement (removal of tissue) offers a 65% probability of curing the infection and saving the foot. If the antibiotics do not stop the infection, there is an 15% probability of death with the remaining 85% surviving and requiring amputation at the knee.

The new, experimental antibiotic has a different set of outcomes. It has a 62.5% probability of curing the infection, with the remainder having amputation at the knee.

With the immediate amputation of the foot option, the outcome is assumed to be certain: the patient will survive. With immediate amputation of the foot, or later amputation at the knee, a prosthesis will be fitted and physical therapy will be required.

Here are the component costs that are used to calculate total cost for each scenario:

parameter	value
Regular antibiotic	$500
New antibiotic	$500
Inpatient cost per day *	$1,000
Foot amputation	$5,000
Foot prosthesis	$2,000
Loss of foot, physical therapy	$10,000
Leg amputation	$10,000
Leg prosthesis	$10,000
Loss of leg, physical therapy	$25,000
Life saving measures	$10,000

(* To calculate the basic cost of the hospital stay associated with the various scenarios, the number of inpatient days for each is also estimated and multiplied by the per diem cost.)

The numeric cost payoffs specified for each outcome in the Cost Formula tree have been hand-calculated using a formula combining the appropriate costs for each particular scenario, including costs of hospitalization, drugs, surgery, prosthetics, physical therapy, and other care.

As reflected in the rolled back tree, shown on a previous page, the least costly option is to treat with the standard antibiotic. A Ranking analysis can be used to calculate the additional cost per patient of the experimental antibiotic, $3427 (or approximately $70000 for each death that is averted).

Implementing a cost formula using variables

The baseline information may be useful in a budget impact calculations. However, you may want to analyze the model on the basis of different estimates of component costs, some of which are uncertain. To do this kind of analysis manually would be tedious, so you decide to implement the cost calculations using a cost formula and perform sensitivity analysis on the component variables.

Start by making a copy of the tree.

- ▷ Open the tree called **Cost Formula**, found in TreeAge Pro's Tutorial Examples subdirectory.
- ▷ Save a copy of the tree, changing the name to **Cost Variables**.

To save some time, a list of definitions of the cost component variables has already been pasted into the tree, at the root node. To see these definitions, turn on the display of variable definitions in the tree.

- ▷ Choose **Edit >** Preferences…, and select the Variables/Markov Info category.
- ▷ Select Show Definitions, and the choose either Expand node… or Wrap at…, so that the full definitions will be visible. Click OK to return to the tree.

Much of the work required to make the tree more flexible has already been accomplished. All necessary variables have been created in the tree, and each has at least one definition.

The remaining tasks to be done in the tree are:

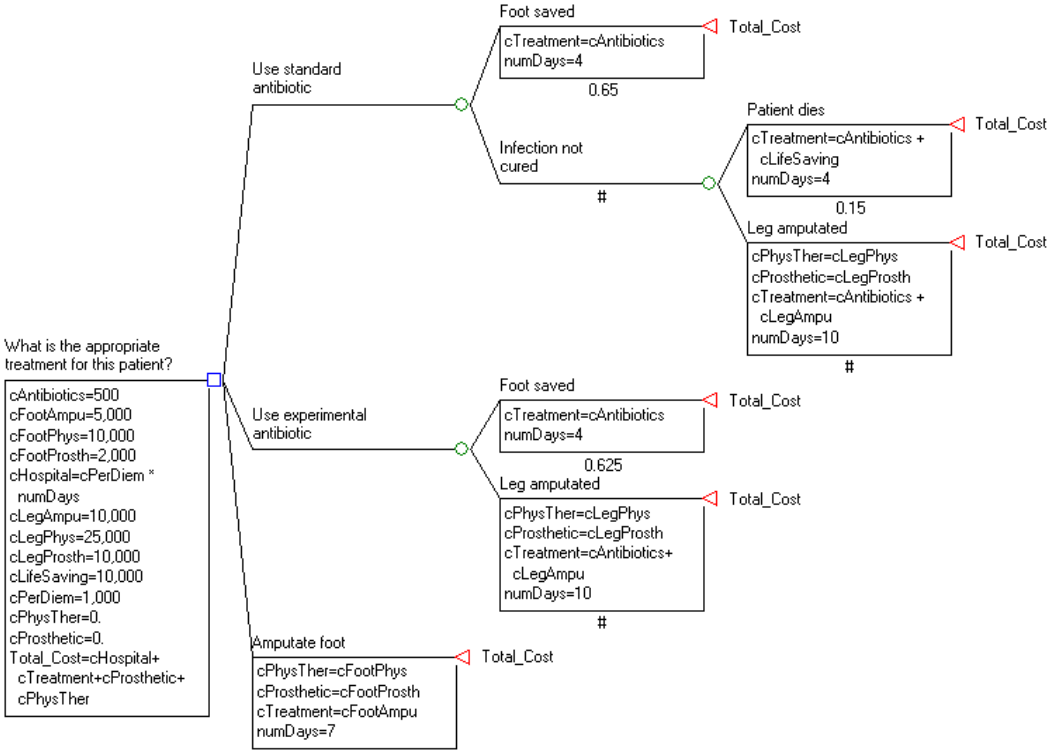
- assign Total_Cost as the payoff of every terminal node
- redefine the *variable* (i.e., not fixed) components of Total_Cost for each outcome, as necessary

The tree below illustrates how and where to add definitions of the variables used in the Total_Cost formula (other solutions are possible, as will be seen later in this chapter). Compare the tree you have opened to the picture of the completed tree.

Variables with multiple definitions

Note that, in the solution, several variables have *multiple definitions*. However, with the exception of numDays, no single variable has more than one *numeric definition*. As a general rule, it is advisable to avoid giving a single variable more than one numeric definition – for example, to represent the probability of two or more distinct events. It is important to follow this rule if you want to be able to perform sensitivity analysis on the variable in question.

(Note: The quantity represented by numDays, used in calculating cHospital, is not intended to be used in sensitivity analysis. If it were, separate variables would be used for each scenario.)



Start by assigning the Total_Cost variable to all of the payoffs. Instead of deleting the existing numeric cost payoffs that are found in payoff #1, however, switch to payoff #2 and assign Total_Cost there.

- ▷ Choose **Edit >** Preferences…. In the Calculation Method category, change Simple calculations to use payoff #2. Click OK to return to the tree.

All terminal nodes should now indicate that no payoff has been assigned. Now, update all the terminal nodes to use Total_Cost as the formula for payoff #2.

- ▷ Choose **Options >** Select If..., select the node type category, and Terminal (or, position in tree = right-most). Click OK. (See Chapter 8 for more on the Select If dialog.)
- ▷ With all terminal nodes selected, choose **Values >** Change Active Payoff (2)…. For payoff #2, type Total_Cost (or select it from the Variables list). Click OK to update the tree.

At this point, although the tree can calculate without causing errors, it will not calculate costs correctly because most of the components of Total_Cost are still defined equal to 0.

> ***Tip:***
> There are two important exceptions to the right-to-left search rule. One involves the recursive definitions of variables, which will be covered at the end of this chapter. The other exception involves the evaluation of tracker variables; refer to the Healthcare Module documentation for details.

Using node-specific variable definitions

Examine the tree to see where and how variables are currently used and defined in the tree, and where definitions need to be added or modified.

To calculate the payoff of any node in the tree, TreeAge Pro must evaluate Total_Cost. To do this, it will search start searching for a definition of Total_Cost at the terminal node, and work leftward until it finds a definition. In this case, the first definition it will find is the default definition of Total_Cost from the root node, which accumulates the following components:

(cost of inpatient care) + (cost of treatment) + (cost of prosthetic) + (cost of physical therapy)

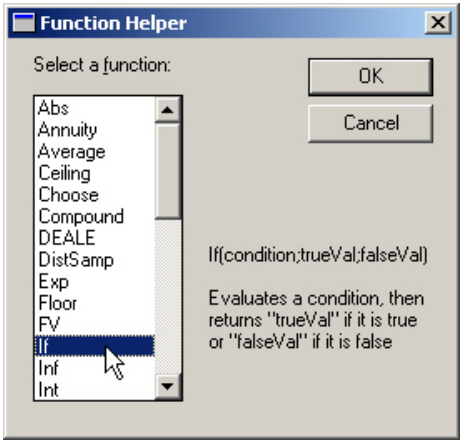
The component variables cHospital, cTreatment, cProsthetic and cPhysTher need to be evaluated. TreeAge Pro restarts the search for each variable's definition *at the terminal node* which is being calculated. This right-to-left search is restarted for each variable encountered in the calculation. (The same process occurs for probability formulas, with a search rooted at the branch being calculated.)

Therefore, for every path in the tree, each variable in the cost formula must have an appropriate definition. Note that for some outcomes, some components do not apply. For example, there are three terminal nodes which represent non-amputation scenarios, and which therefore have no prosthetic or therapy costs. These components should have 0 values in these paths.

You should confirm that, when rolled back, the Cost Variables tree provides the same results as the original Hospital Cost tree.

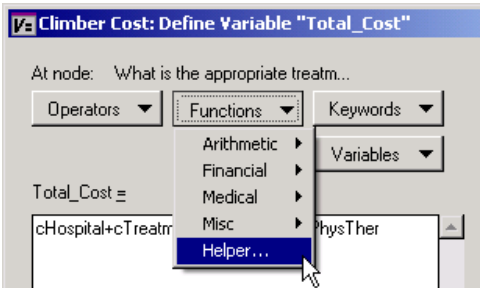
Using mathematical, statistical, and other functions

TreeAge Pro includes a wide variety of functions which make it easier to perform some commonly-used mathemetical operations, or in some cases to provide access to special features in TreeAge Pro.



Notes on using functions in formulas

- TreeAge Pro's built-in functions are case-insensitive, with the name followed by parentheses, and in most cases take arguments in the parentheses.
- Functions that have multiple arguments must use semicolon (";") separators.
- In most cases, any valid expression can be used as an function argument (with the exception of MatrixMult).
- In the tree window, use the Function Helper (see Chapters 13 and 14) for help in assigning the correct parameters to each functions.



- Functions which here indicate an argument named "LIST" take a flexible number of arguments. For example, Average( ) returns the arithmetic mean of all of its arguments, so Average(1;4;8;13) = 6.5.
- To test a function, use the Calculator/Evaluator tool described in Chapter 13
- Chapter 17 covers a special set of functions used in Monte Carlo simulation.
- Chapter 21 covers a special set of functions that take text string arguments, instead of numeric or variable expressions.
- Chapter 35 covers a special set of functions used in Markov modeling, for example in calculating transition probabilities.

Arithmetic functions

In the arithmetic functions, "e" represents the base of the natural logarithm (approximately 2.718.)

Function	Explanation
Abs(x)	absolute value of x
Average(LIST)	arithmetic mean of a list
Ceiling(x)	smallest integer larger than x
Exp(x)	"e" to the xth power
Floor(x)	greatest integer smaller than x
GammaFn(x)	n! for integers less than 19; Stirling's approximation otherwise
Int(x)	integer component of x
Ln(x)	natural (base "e") logarithm of x
Log(x)	base 10 logarithm of x
Max(LIST)	maximum, or highest value, of a list
Min(LIST)	minimum, or lowest value, of a list
Modulo(x; y)	remainder of x divided by y
Prod(LIST)	product of a list
Root(x; y)	y^{th} root of x
Round(x)	x rounded to the nearest integer
Sqrt(x)	square root of x
Stdev(LIST)	standard deviation of a list of numbers.
Sum(LIST)	sum of a list

Financial/discounting functions

In functions, a "rate" argument can be entered as either a percent or a decimal. For example, eight percent can be represented as either "8%" or ".08," but not simply as "8."

Function	**Explanation**
Annuity(rate; #periods)	To calculate the net present value of a series of equal future payments, multiply this function times the amount of a single payment. *Formula:* $(1-(1+rate)^{-periods})/rate$
Compound(rate; #periods)	Returns the compound interest rate (effective yield) at a fixed rate over a fixed number of periods. If used as a multiplier, this function can be used to calculate future value; if used as a divisor, it can be used to calculate discounted, present value. *Formula:* $(1+rate)^{periods}$
Discount(util; rate; time)	Discounts a specified value (cost or utility) at the specified discount rate over the specified period. *Formula:* $util / ((1 + rate)^{time})$
FV(pmt; pv; rate; #payments)	Returns the future value of a series of equal, periodic payments. The "pv" parameter represents an initial payment. *Formula:* $[pmt*((1+rate)^{payments}-1)]/rate + pv*(1+rate)^{payments}$
NPV(rate; LIST of flows)	Returns the net present value of periodic cash flows, discounted. *Formula:* $\sum(f_i / (1+rate)^i)$
PMT(principal; rate; term)	Returns the size of equal, periodic payments required to pay off a loan, given the principal, interest rate, and term of the loan. *Formula:* $principal*(rate/(1-(1+rate)^{-term}))$
PV(pmt; fv; rate; #payments)	Returns the present value of equal, periodic payments at a fixed interest rate. The "fv" parameter represents a final payment. *Formula:* $[pmt*(1-(1+rate)^{-payments})]/rate + fv/((1+rate)^{payments})$
UtilDiscount()	Obsolete. See Discount(), above.

Miscellaneous functions

Function	Explanation
Bilink(index)	Returns the value associated with a dynamic link. See Chapter 20.
BranchProb()	Returns the calculated branch probability for the node being calculated.
Choose(index;LIST of values)	Returns a value based on its location in the LIST, as specified by the index. For example, "Choose(2;100;200;300)" returns 200, because the index is 2, and 200 is the second value in the list. The index must be a positive integer; an error is reported for fractional or out-of-range values.
If(condition; trueval; falseval)	Evaluates a condition and returns "trueval" if the condition is true or "falseval" if the condition is false. For example, "IF(x<0;50;75)" would return 50 if x were negative and 75 if x were non-negative.
Inf()	Returns infinity (∞). Although no arguments are accepted, you must type the left and right parentheses.
Link(index)	Returns the value associated with a DDE link. See Bilink() above, and Chapter 20.
Pi()	Returns π, or approximately 3.1416.
PathProb()	Returns the cumulative path probability for the node being calculated (the Markov node in a Markov subtree).
StateProb(A; B)	Returns the state probability of one or more states at the start of the current cycle. See Chapter 35 for details on this Markov modeling function.
Sub(index)	Obsolete. See Bilink() and Link(), above.

Choose() function tip:

Performing a sensitivity analysis on a variable used as the index of the *Choose()* function, from 1 to the number of list items, can show the impact on calculations of the different values specified in the list.

Nesting functions:

All functions can be *nested*. The If( ) function is frequently used nested within another If( ) function call. This is because the If( ) function tests for one condition, and can return either of two values, normally. However, by using another If( ) function as one of the return values, it is possible to perform two tests and return any of three values (and so on).

String functions ...

Chapter 21 covers a special set of functions that take text string arguments, instead of numeric or variable expressions.

Markov functions ...

Chapter 35 covers a special set of functions used in Markov modeling, for example in calculating transition probabilities.

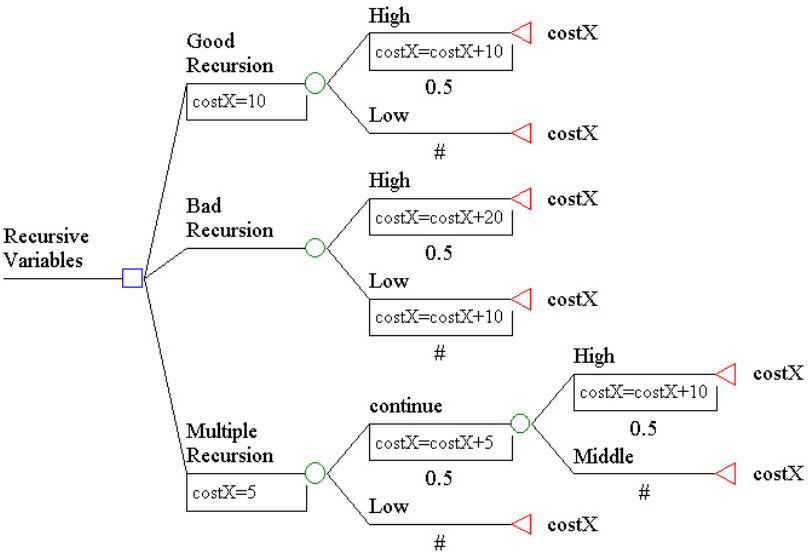
Recursive variable definitions

A recursive variable definition means a definition in which the variable being defined also occurs *in* the definition. Recursive definitions can be an effective way to build complex cost formulas, like those in the Cost Variables tree. Rather than creating one or more long formulas, you can gradually build the formula, adding components as events occur in each particular scenario.

How recursive definitions work

During calculation of a probability, payoff, or Markov reward, when the standard right-to-left search for a definition of a particular variable (e.g., "costX") first locates a definition (e.g., "costX=1,000"), TreeAge Pro stops looking for additional definitions of that variable. However, when the first definition TreeAge Pro encounters in the search is a self-referential, *recursive* definition (e.g., "costX=costX+1,000"), the variable (e.g., "costX") is flagged as a recursive variable, and TreeAge Pro continues searching for additional definitions of that variable to the left of the node where the first, recursive definition was found.

Open the model called Recursive Variables, shown below, to see an example of this process.



All payoffs in the tree reference the same variable, costX, which has no default definition. The decision node's topmost subtree, labeled Good Recursion, illustrates a valid recursive definition. When calculating the payoff of the first terminal node in the Good Recursion subtree, labeled High, the normal, right-to-left search for a definition of costX finds the self-referential definition costX=costX+10 at that terminal node.

For the purposes of the current terminal node payoff calculation, costX is now identified as a recursively defined variable. The search for additional definitions of costX is now continued one node to the left, at the Good Recursion node. There, the non-recursive definition costX=10 is found, the search is complete, and the payoff calculation can be carried out.

Select the terminal node labeled High in the Good Recursion subtree, and choose Analysis > Expected Value. The calculated value is 20.

A non-recursive definition of the recursive variable, like costX=10 at the Good Recursion node, must eventually be found; definitions can't be infinitely recursive.

Look at the second branch of the decision node, labeled Bad Recursion. In this subtree, when either terminal node's payoff is evaluated, TreeAge Pro's variable definition search locates a recursive definition. In both cases, the search for additional definitions is started one node to the left, at the Bad Recursion node. No non-recursive definition of x has been made there or at the root node. Therefore, if you try to calculate the Bad Recursion subtree, an error message will be shown. To see this, select the Bad Recursion node or either of its terminal nodes and choose Analysis > Expected Value command.

Multiple recursions, using a series of recursive definitions of a variable, will work, as shown in the Multiple Recursion subtree. Simply ensure that a numeric definition of the recursive variable will eventually be found. For example, in calculating the payoff of the terminal node labeled Middle in the Multiple Recursion subtree, the variable definition search locates the recursive definitions costX=costX+10 and costX=costX+5, and finally the non-recursive definition costX=0. Thus, the calculated value of the Middle terminal node is 15.

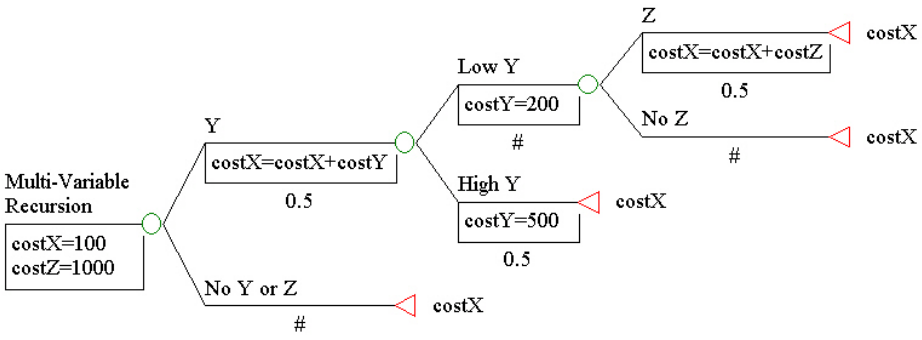
Complex recursion

Other variables may be referenced in a recursive definition. To see how TreeAge Pro's variable definition search works with a combination of recursive and non-recursive variables, open the Multi-Variable Recursion tree from the TreeAge Examples subdirectory. To better explain this complex example, the text will illustrate the incremental changes in the payoff calculation formula during TreeAge Pro's variable definition search.

All payoffs in the Multi Variable Recursion tree use the same variable, costX (payoff = costX). When the terminal node labeled Z is evaluated, the variable definition search finds the recursive definition costX=costX+costZ at the terminal node (payoff = costX + costZ).

Before a recursive search for costX is continued one node to the left, TreeAge Pro looks for a definition for costZ. The variable costZ is evaluated *as a normal variable*, which means that the search for a definition of costZ is started *at the node being calculated*, the terminal node Z. A normal right-to-left search locates the non-recursive definition costZ=1000 at the root node (payoff = costX + 1000).

The recursive search for additional definitions of costX is then continued at the Low Y chance node, which is one node to the left of the node where the initial, recursive definition of costX was found. Another recursive definition, costX=costX+costY, is found at the chance node labeled Y (payoff = [costX + 1000] + costY).



As above, before the recursive search continues, a normal variable definition search for costY is initiated at the terminal node being calculated, Z. At the chance node Low Y, the non-recursive definition costY=200 is located (payoff = [costX + 1000] + 200).

TreeAge Pro proceeds with the search for additional definitions of costX, starting at the root node. A non-recursive definition of costX is required, and found: costX=100. The final payoff formula for node Z, therefore, is [100 + 1000] + 200. Calculating an expected value for the Z terminal node returns the value 1300.

Before using complex recursion in your models, it is important that you be thoroughly familiar with the logic underlying both standard and recursive variable definitions in TreeAge Pro. You are also urged to test your model to make sure it appears to be calculating correctly.

Chapter 15

More Sensitivity Analysis Tools

Chapters 12–14 provided detailed instructions on representing uncertain values using variables and performing one-way sensitivity analysis. This chapter covers sensitivity analysis options, multi-way sensitivity analysis, and special variations on one-way sensitivity analysis.

In this chapter:

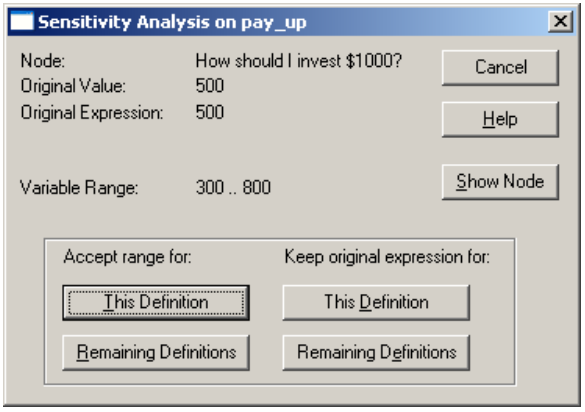
- Analyzing variables with multiple definitions 186
- Tornado diagrams 188
- Two-way sensitivity analysis 191
- Three-way sensitivity analysis 194
- Threshold analysis 195
- Analyzing correlated variables 198
- Additional sensitivity analysis topics 199

Analyzing variables with multiple definitions

While first learning how to use variables, it is not uncommon to unintentionally end up with a variable defined numerically at multiple nodes. While there are good reasons why TreeAge Pro allows the same variable to be defined at multiple nodes — as described in Chapters 12 and 14 — this situation is usually neither necessary or desirable with a variable intended for sensitivity analysis, as it may result in errors.

The tree at the bottom of the page illustrates the potential problem; note the two numeric definitions of pay_up.

If you try to perform a sensitivity analysis on the variable pay_up in this version of the investment tree, TreeAge Pro cannot immediately run the analysis — not without more input, to clarify which definition should vary during the analysis. Therefore, the special dialog shown at right will appear.

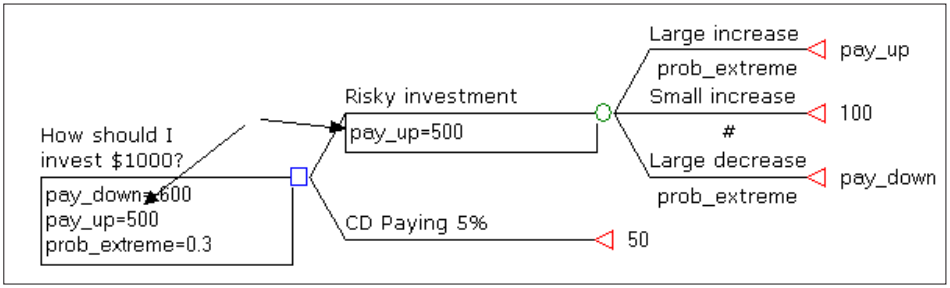


Your goal when learning to use variables for sensitivity analysis should be to avoid ever having to use this dialog, as it may be difficult to avoid making errors in choosing which definitions should be varied in the sensitivity analysis. The solution, in this case, is to delete one of the definitions of pay_up.

> ***Recommendation:***
> If you plan to carry out sensitivity analysis on a variable's value, ensure that the variable has a single numeric definition (e.g., define it only at the root node, or only at one of the decision node's branches).

Following the recommendation on the previous page will help you avoid errors in analyses other than sensitivity analysis, as well.

For example, in the version of the investment tree shown on the previous page, changing the definition of pay_up at the root node will have *no impact on payoff calculations*. TreeAge Pro will use only the definition of a variable that is closest to the node being calculated, a terminal node in this case. This point is critical to understanding how to use variables in TreeAge Pro.



Concept:
When TreeAge Pro calculates a payoff or probability that references a variable, it looks for a definition of the variable *starting at the node being calculated* and, if necessary, moving to the left (i.e., toward the root node). TreeAge Pro uses the first definition found during this *right-to-left* search.

For a variable used in a payoff, TreeAge Pro looks for a definition starting at the terminal node. In the case of a variable used in a branch's probability, the search for a definition begins at that branch.

Here is another simple but instructive example based on the investment tree, this time using variables for Risky investment's payoffs. Again, there is a problem with a variable having multiple numeric definitions.

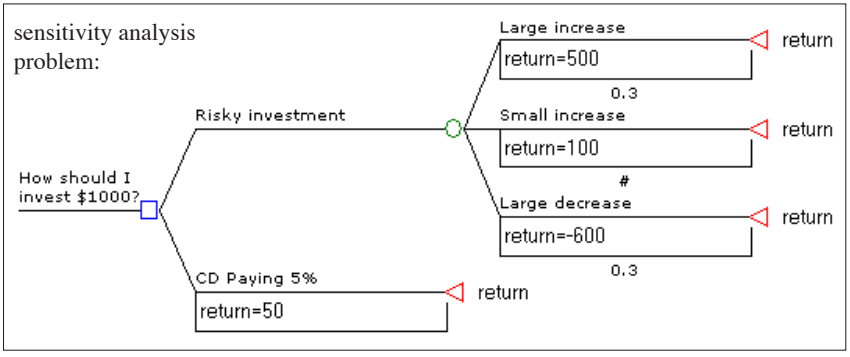
While the version of the investment tree shown below right will roll back correctly, it is not well-formed for sensitivity analysis. The variable "return" is defined numerically four times, once for each scenario. Performing a sensitivity analysis will not be straightforward.

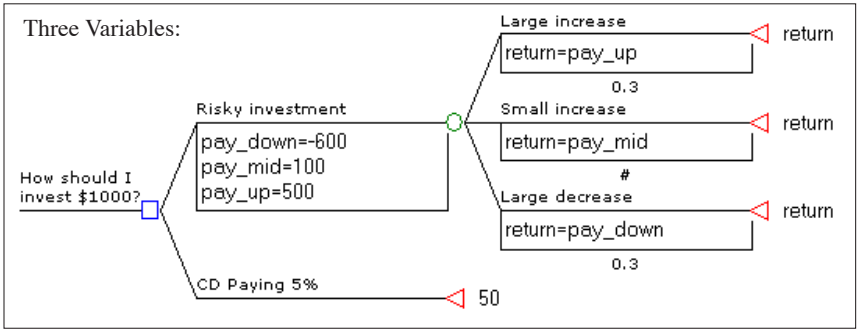
A better design would be like that used in the Three Variables version of the tree, with different variables for each uncertain payoff. This tree is shown at the very bottom of the page

If it were necessary or desirable to define all payoffs using the same variable, another option would be to *not* define return numerically. Instead, it could be defined equal to other variables, as shown below.

In this case, the three numerically-defined variables would be used for sensitivity analysis.

More details on defining variables in terms of other variables is provided in Chapter 14.





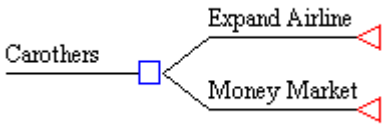
Tornado diagrams

A tornado diagram is a set of one-way sensitivity analyses brought together in a single graph. It can include any number of the variables defined in the tree.

In the graph, a horizontal bar is generated for each variable being analyzed. Expected (or incremental) value is displayed on the horizontal axis, so each bar represents the selected node's range of expected (or incremental) values generated by varying the related variable. A wide bar indicates that the associated variable has a large potential effect on the expected value of your model.

The graph is called a tornado diagram because the bars are arranged in order, with the widest bar (potentially the most critical uncertainty) at the top and the narrowest bar at the bottom, resulting in a funnel-like appearance.

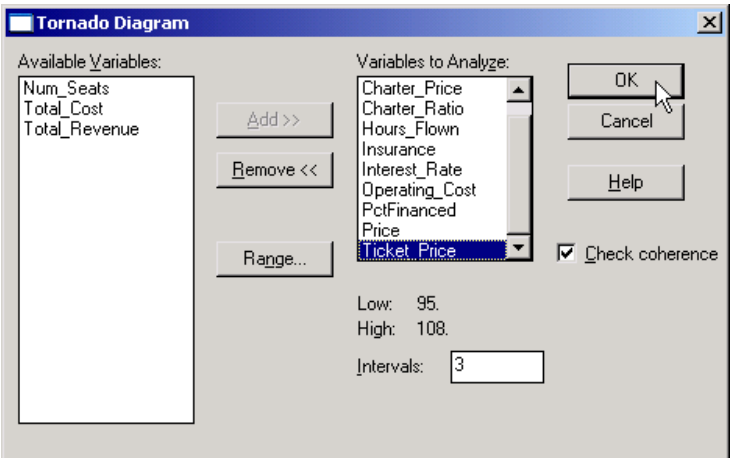
The example tree Airline Problem is ready for a tornado diagram. The model is a simple cost function, each of whose inputs may be varied to see how each may affect the expected value.



► **To create a tornado diagram:**

▷ Select the decision node and choose **Analysis >** Sensitivity > Tornado Diagram.

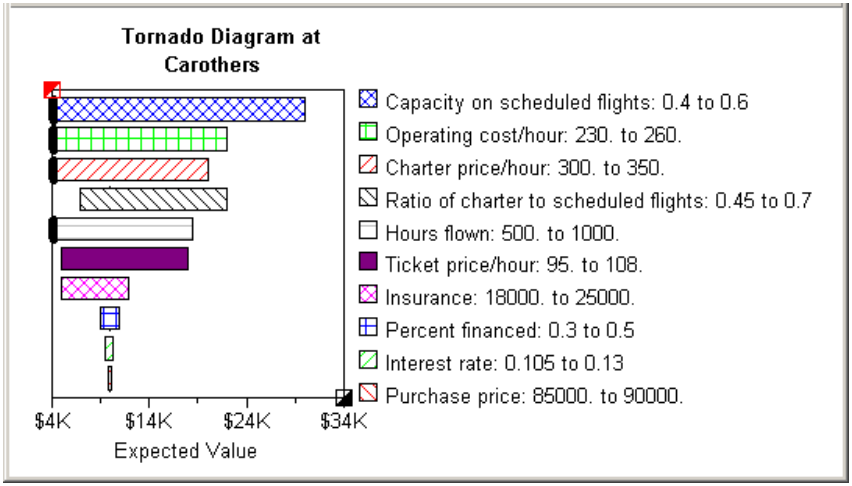
▷ In the Tornado Diagram setup dialog, select each variable to analyze and click the **Add >>** button. This will move the variable to the list on the right, and prompt you for a range and number of intervals for that variable.



▷ Be careful not to select variables that are formulas (Total_Cost, Total_Revenue, Operating_Cost), rather than numeric parameters

▷ Click OK to start the analysis.

The tornado diagram for the airline decision is shown on the next page.


Tornado Diagram at Carothers
Capacity on scheduled flights: 0.4 to 0.6
Operating cost/hour: 230. to 260.
Charter price/hour: 300. to 350.
Ratio of charter to scheduled flights: 0.45 to 0.7
Hours flown: 500. to 1000.
Ticket price/hour: 95. to 108.
Insurance: 18000. to 25000.
Percent financed: 0.3 to 0.5
Interest rate: 0.105 to 0.13
Purchase price: 85000. to 90000.
$4K
$14K
$24K
$34K
Expected Value

Each bar represents a one-way sensitivity analysis performed at the selected node. The tornado diagram includes a vertical dotted line indicating the expected value. You can use this as a visual fulcrum, to view the impact of each variable relative to the original (baseline) expected value.

If you click any bar once and hold down the mouse button, you will see the input and output range for that parameter. The *input* range is the range over which you varied the associated variable. The *output* range is the resulting range of expected values when the variable is varied.

If you double-click on a bar, you will see the full line graph as it was generated from the sensitivity analysis. All relevant threshold information will be included.

Tornado diagrams can be created at chance nodes and decision nodes. At a decision node, any threshold (i.e., change in policy) found will be identified in a variable's tornado bar with a heavy vertical line. Threshold lines are drawn at the expected value on the x-axis at which the optimal path changes.

If a threshold appears at either end of a bar, this usually indicates that an alternative which is optimal for part of the analysis range has an unchanging expected value in that range. To see details about the change in policy associated with a threshold line, double-click the bar to view the full line graph for the variable.

You may turn off the display of individual bars in the analysis. Select Graph > Show/Hide (when the graph window is in front) to indicate which bars should be displayed.

> ***Tip:***
> Because setting up the tornado diagram analysis can be time consuming, you may want to store the setup information, using the Analysis > Storage > Save Last… command. See Chapter 19 for details on using stored analyses.

Incremental calculations

A tornado diagram can report, instead, the sensitivity of the incremental/marginal value calculated between two strategies.

► **To create an incremental tornado diagram:**

▷ Instead of selecting a decision node, select two of its branches. To do this, click on one branch and then shift-click on a second branch. Then choose **Analysis >** Sensitivity > Tornado Diagram.

Additional calculations in the text report

The tornado diagram's text report will display, in addition to the input and output ranges for each parameter, a number of other useful calculated values.

- **Spread** – This is the width of the bar (i.e., High EV - Low EV).
- **SpreadSqr** – The spread value, squared.

Adding the SpreadSqr values to calculate a net risk value, two additional measures of uncertainty are then calculated for each variable.

- **Risk Pct** – This is a measure of how much of the total uncertainty is represented by the specified bar (equals SpreadSqr / NetRisk). The RiskPct values sum to 1.0.
- **Cum Pct** – A cumulative version of Risk Pct., making it easy to scan the bars and say "to address 90% of the risk, I must consider the uncertainty represented by the following variables…."

Including correlated variables in the tornado diagram

As in all sensitivity analyses, if you select a variable for the tornado diagram which has correlations, you have the option of including the correlated variables in the analysis, as well. Correlates which are varied together in the analysis will appear as a single bar, for which you provide a description (e.g., "Pharmacy Costs").

The names of the correlated variables in a given group will not be displayed in the tornado diagram itself. If you single-click the bar representing those parameters, the names and input ranges of all correlated variables in the group will be shown.

Details on how to correlate variables is provided later in this chapter.

Variable	Low EV	High EV	Low Input	High Input	Spread	SpreadSqr	Risk Pct	Cum Pct
Capacity on scheduled flights	$4K	$30K	0.4	0.6	25775	664350625	35.322839956	35.322839956
Operating cost/hour	$4K	$22K	230.	260.	17775	315950625	16.798770018	52.121609974
Charter price/hour	$4K	$20K	300.	350.	15775	248850625	13.231131979	65.352741953
Ratio of charter to scheduled ...	$7K	$22K	0.45	0.7	15000	225000000	11.96301876	77.315760713
Hours flown	$4K	$18K	500.	1000.	14275	203775625	10.834540554	88.150301268
Ticket price/hour	$5K	$18K	95.	108.	13000	169000000	8.985556313	97.135857581
Insurance	$5K	$12K	18000.	25000.	7000	49000000	2.605279641	99.741137222
Percent financed	$9K	$11K	0.3	0.5	2012.5	4050156....	0.215342645	99.956479867
Interest rate	$9K	$10K	0.105	0.13	875	765625	0.040707494	99.997187361
Purchase price	$10K	$10K	85000.	90000.	230	52900	0.002812639	100

Two-way sensitivity analysis

Two-way sensitivity analysis is used to examine the impact on a decision of simultaneous changes in the values of two variables. One method is to run a series of one-way analyses, each time incrementing the value of a second variable; the resulting line graphs can be merged (see Chapter 6).

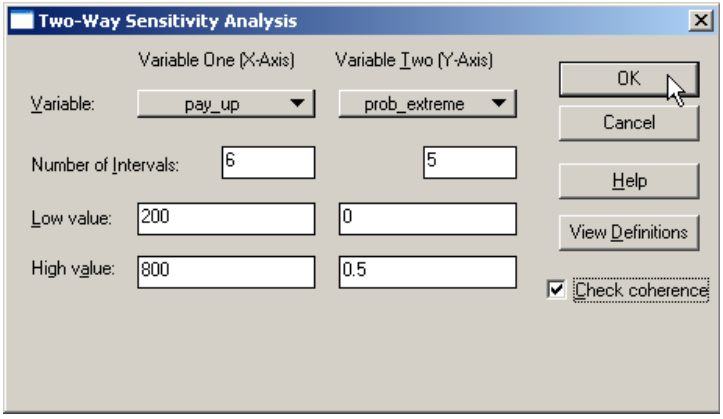
Another method available in TreeAge Pro is to automate this series of analyses and present the results in a region graph. The region graph very efficiently identifies changes in the optimal policy as the values of the two variables change.

The two-way sensitivity analysis setup dialog resembles the one-way dialog, except that you must specify two variables and a range of values for each. A decision node must be selected in order to perform a two-way sensitivity analysis.

► **To perform a two-way sensitivity analysis:**

▷ Open the file you created in Chapter 12 called Three Variables (or the copy of this file called Three Vars, found in TreeAge Pro's Tutorial Example subdirectory).

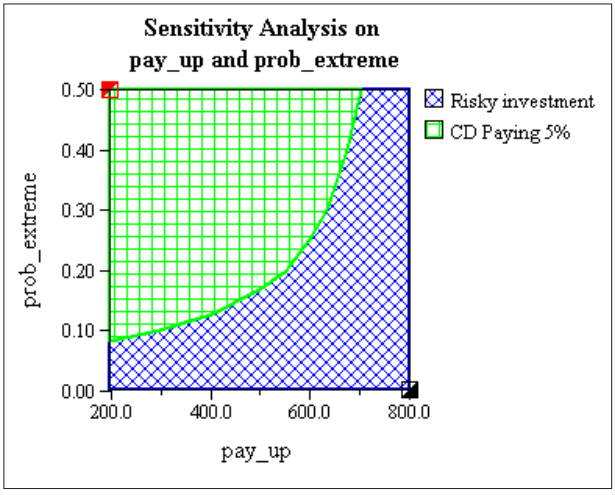
▷ Select the root node, and choose **Analysis** > Sensitivity > Two-Way.



▷ For one variable, select pay_up, and specify 6 intervals in the range from 200 to 800.

▷ For the second variable, select prob_extreme, and specify 5 intervals in the range from 0 to 0.5. Click OK to run the simulation.

The resulting graph is shown at left. It identifies which strategy is optimal in regions of values of the variables; thresholds are simply the border between two regions.



Tip:

Use two-way analysis *only* when the two variables are independent. If the two variables are a correlated pair, be sure to turn off the correlation. A legend displaying the name and the range of any correlated variable involved in the two-way sensitivity analysis will be placed adjacent to the appropriate axis.

Choosing intervals

Based on the selecting X intervals for one variable, and Y intervals for the other, TreeAge Pro recalculates the tree (X+1)*(Y+1) times, for the different combinations of values of the two variables. In the example on the previous page, 42 (7x6) recalculations occur, creating a grid of sensitivity analysis results.

You may find that a two-way analysis requires more intervals per variable to attain a reasonable level of accuracy than a one-way analysis. This is because a two-way analysis graphically represents only the threshold values – the optimal path crossings. The one-way analysis may show significant details which are simply not shown in the two-way analysis.

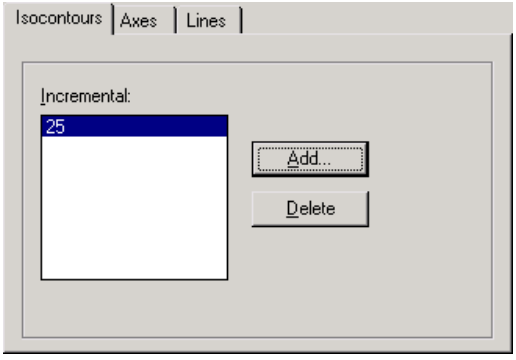
The graphical representation of the results of two-way sensitivity analysis has some unavoidable limitations; this also applies to three-way analysis.

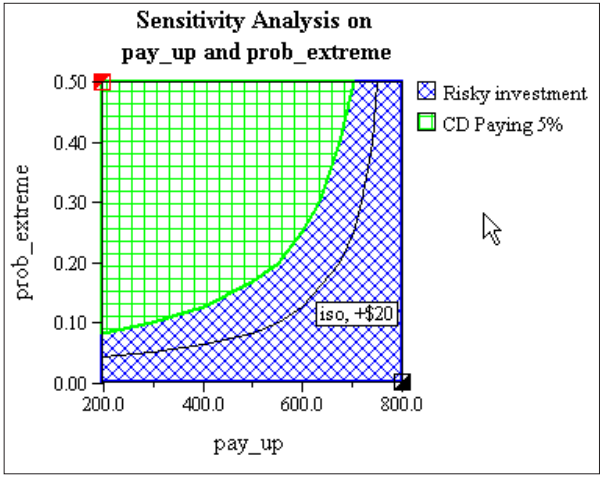
- The accuracy of threshold lines may be compromised around the edges of the graph. The unavoidable result of using approximation techniques to identify thresholds is the appearance of distortion when two edges of a region of optimality draw closer together than one-half the width of an axis interval. Accuracy can be enhanced by running the analysis using more intervals.
- Regions of indifference are not shown. Areas of the graph where indifference exists are, instead, assigned to one of the decision alternatives. You should use the text report (accessed via the **Actions >** or **Graph >** Text Report… command) to identify any areas of indifference by comparing the expected values at each interval.

Isocontours

Isocontour lines can be setup in *any* two-way sensitivity analysis graph comparing two alternatives. An isocontour represents, for the combinations of variable values along the line, a constant marginal or incremental value of the *topmost branch of the decision node*. (A threshold line between different regions in the graph, if it exists, is actually an isocontour showing where the incremental value is zero (i.e., where you are indifferent between the alternatives).

To add custom isocontours, click the Options button or select Graph > Options while a two-way sensitivity graph (comparing two options) is active. Select the Isocontours tab, and enter the incremental values to use for isocontours.



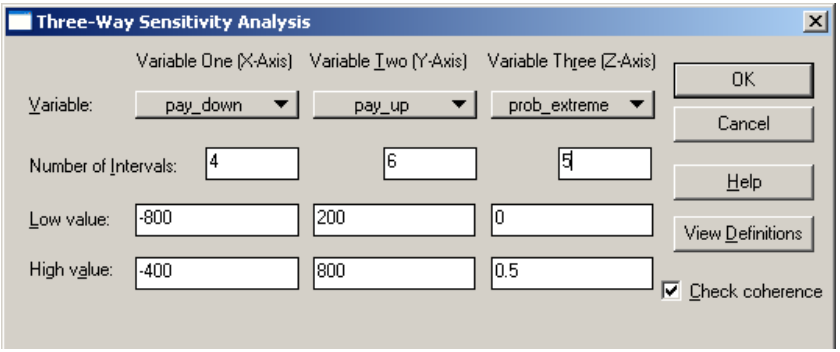


Labels are not automatically created for isocontours; label the lines manually, using the **Actions >** New Label… command described in Chapter 6.

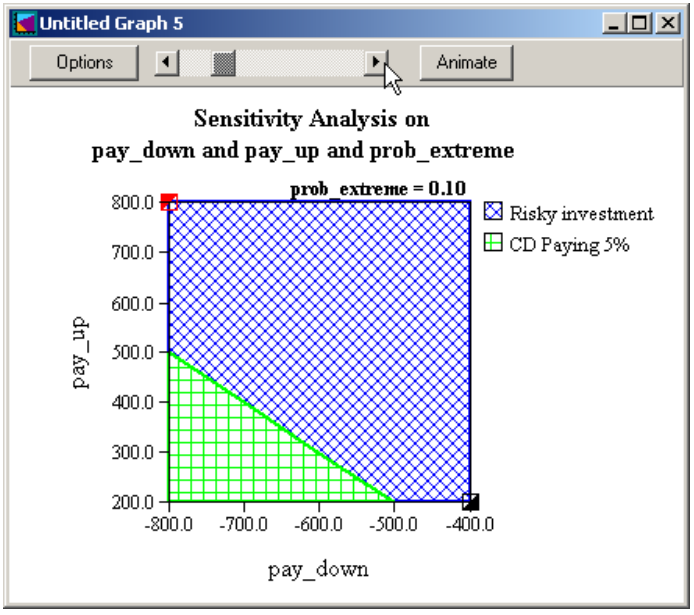
> ***Tip:***
> Adding only a 1000 isocontour, will *not* show lines representing both +1000 and -1000 incremental values; you need to add both 1000 and -1000 values to the list of isocontours in the graph to see both the +/- 1000 isocontours in the graph. Negative value isocontours, in this context, merely indicate that the comparator has changed from the topmost option to the bottom branch.

Three-way sensitivity analysis

The three-way sensitivity analysis dialog looks similar to the two- and one-way sensitivity analysis dialog.



The results of a three-way sensitivity analysis are presented as an animated two-way sensitivity analysis region graph. The third variable is represented not with its own axis, but rather using a series of two-way graphs — if four intervals are specified for the third variable, then five graphs will be created, and shown in series.



Use the Animate button or the scroll bar to cause the third variable to cycle through its range, interval by interval. At each frame you will see a snapshot of the three-way analysis, showing you how the two-way analysis of the first two variables is affected by varying the value of the third variable. At each frame, the value of the third variable is displayed in a special label near the top-right corner of the graph.

Threshold analysis

This specialized form of sensitivity analysis offers the ability to search more thoroughly and accurately for threshold information. The result of this analysis is a detailed, textual description of how the optimal strategy is affected by changing the value of a single variable across a designated range.

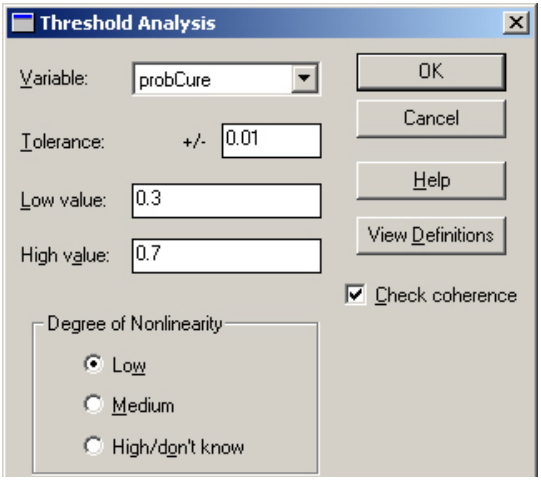
In a standard one-way sensitivity analysis, the user designates the number of intervals into which the range is to be divided; actual calculations occur only at these intervals. As a result, the accuracy of the associated threshold analysis is limited to values determined by linear interpolation.

In contrast, threshold analysis has been designed to maximize accuracy of the analysis in situations where accuracy is more critical than speed. The specified range is iteratively searched until a specified minimum tolerance is reached.

How threshold analysis works

After you select Threshold Analysis, a dialog box will appear asking four questions.

- the name of the variable on which the sensitivity analysis is to be performed;
- the value range over which to vary the designated variable;
- a value for the “threshold tolerance”; and
- estimate the non-linearity of the model.

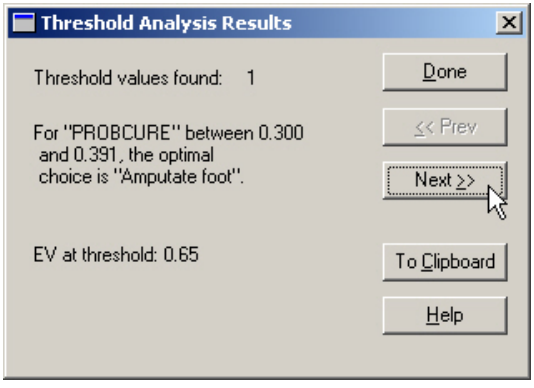


Understanding the results

A dialog box will appear with the results of the threshold analysis. It will specify the number of threshold values identified by the analysis, the optimal policy between the low end of the range and the first threshold (change in policy), and the expected value at the threshold.

The dialog box specifies a single interval throughout which the optimal policy is consistent. The term “EV at threshold” refers to the expected value when the variable in question is given the value at the top of the interval being described. The “Prev” and “Next” buttons can be used to view each of the other policy intervals.

Use the To Clipboard button is used to copy the threshold report to the clipboard.



Threshold tolerance

The tolerance is stated in the same units of value as the variable in question; it is not a percentage. The tolerance is related to the value of the variable, not to expected value. Thus, entering a tolerance of 0.1 means that the actual location of any threshold will be within plus or minus 0.1. For example, if TreeAge Pro indicates finding a threshold at Var=0.391, this means that the threshold definitely occurs somewhere between 0.381 and 0.401. Because TreeAge Pro applies linear interpolation after it meets your tolerance, you can expect the actual reported value to be even more accurate than the tolerance.

The designated tolerance has a second function. TreeAge Pro uses this value as a basis for determining the number of decimal places (not significant digits) to specify in the result. The number of decimal places displayed will be one greater than the number of decimal places specified in the tolerance. Assume, for example, that TreeAge Pro finds a threshold at 0.459033333. If the tolerance had been set at 0.1, the threshold value would be reported as 0.46. If, instead, the tolerance were set at 0.01, the threshold would be displayed as 0.459.

Threshold analysis non-linearity

A sensitivity analysis may indicate multiple threshold values. However, this series of changes in policy will be identified correctly by TreeAge Pro only if the thresholds appear in different intervals in the first iterative pass. Since linear interpolation is used to find thresholds in a sensitivity analysis, *only one threshold can be found per analysis interval*.

For example, suppose that two thresholds exist in the same analysis interval, with optimality switching between the same decision options. Since the same policy is optimal at both ends, notwithstanding the intervening thresholds, TreeAge Pro will assume that no thresholds occur in that interval. There is no way to avoid this problem entirely. TreeAge Pro could subdivide a range into 100 intervals and still miss policy changes within an interval if the same optimal policy is specified at both ends. Even if different strategies are optimal at either end of an interval, and TreeAge Pro identifies a threshold in that interval, it is still possible that one or more additional thresholds in that same interval will have been missed.

For example, three alternatives, A, B, and C, might be compared using a sensitivity analysis; A is optimal at the beginning of an interval, B in the middle, and C at the end. Although you know that two thresholds (A to B, then B to C) actually occur, TreeAge Pro will find just one (a nonexistent one, A to C) from looking at the optimal alternative at the ends of the interval.

The non-linearity hint is an attempt to minimize the likelihood this will occur. The more non-linear you describe the graph to be, the smaller the interval used by TreeAge Pro, so as to ensure catching any double thresholds.

Performing a one-way sensitivity analysis on the variable in question before performing a threshold analysis will indicate the appropriate non-linearity setting. If the sensitivity analysis graph is very nonlinear (i.e., it has multiple thresholds), use a higher setting for the measure of non-linearity. This will cause TreeAge Pro to increase the number of intervals searched at each step. Increasing the non-linearity setting also increases the time needed to perform the analysis. For this reason, it is not recommended that you automatically use the Medium or High settings.

Initially, TreeAge Pro will subdivide the given range into a number of intervals. The number of intervals searched relates to the non-linearity radio buttons as follows:

- Low: 4 intervals
- Medium: 8 intervals
- High/Don't Know: 12 intervals.

At each interval where a change in optimal strategy is identified, TreeAge Pro will either

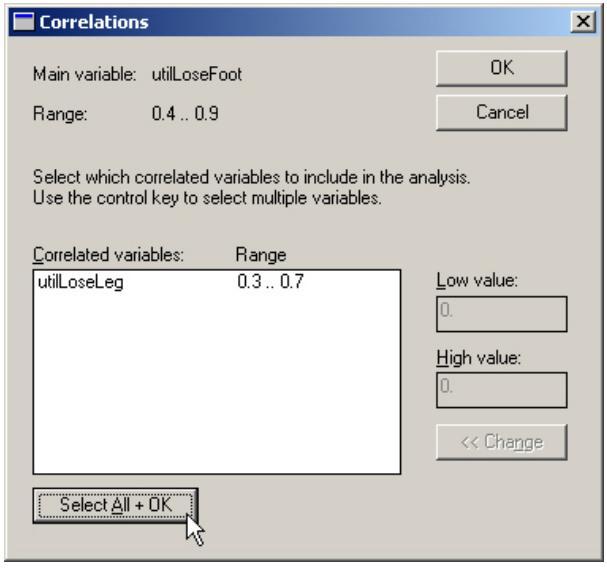
- calculate a threshold value, if the width of the range is less than twice the given tolerance; or
- redivide the interval into 4, 8, or 12 subintervals, as indicated, and search those, as above.

As the calculation proceeds, the progress bar shows how far over the given range TreeAge Pro has searched. If a threshold value is found, the bar will slow down considerably, but it will move more quickly over intervals in which no threshold value is found.

Analyzing correlated variables

Linkages (i.e., perfect positive or negative correlations) between pairs of variables can be specified, for use during sensitivity analysis. These value linkages are set up in the Properties dialog box (see the section Variables and Table) for either member of a correlated pair, by clicking on the **Correlations...** button. (See Chapter 12 for instructions on defining the sensitivity analysis and other properties for a variable.)

Once created, the correlation is identified in the properties of *both* variables. Thus, when you choose to perform a sensitivity analysis on either member of a correlated pair of variables, TreeAge Pro will remind you that one of the parameters to be varied has one or more correlates that can also be varied during the analysis. Once the analysis parameters have been entered, you will have the option of simultaneously varying any or all correlated variables over their own value ranges.



► **To perform a sensitivity analysis using correlated variables:**

▷ Open the tree called Correlated Variables, found in the Tutorial Examples subdirectory. This tree has a positive correlation specified between the variables utilLoseFoot and utilLoseLeg.

▷ Select the root, decision node and choose **Analysis >** Sensitivity Analysis > One-Way….

▷ In the Sensitivity Analysis dialog, select the variable utilLoseFoot (or utilLoseLeg). Specify `10` intervals and a range of `0.4` to `0.9`. Click OK.

After choosing to perform a sensitivity analysis on a variable with correlations, the Correlations dialog (shown below) will appear. From this dialog, it is possible to specify which correlations should be active during the analysis.

To include all displayed correlations, simply click the Select All + OK button. To include only particular correlated variables, highlight each variable name for inclusion and then click OK. Finally, to exclude all correlates, click the OK button immediately on entering the Correlations dialog, leaving all correlate variable names unselected.

It is also possible to change the range of values applied to each correlated variable. Clicking on a correlate's name in the list activates text boxes allowing changes to the default high and low values. To update the range using the changes you make, click on the Change button below the two text entry boxes. During the analysis, the same number of intervals specified for the original variable will be used in dividing the range applied to each correlated variable.

Setting ranges for correlated variable

Be sure to pay close attention to the ranges for correlated variables. Because TreeAge Pro does not parameterize the correlation, you must ensure that if you use a narrow range for one variable (its 75% confidence intervals, for example), you also use an appropriately narrow range for the correlated variables.

It is not possible to change the type of correlation (positive or negative) from within the sensitivity analysis dialog; these changes must be made in the variable's Properties dialog. Detailed instructions on modifying the sensitivity analysis properties of variables can be found in Chapter 12.

Graphs generated for sensitivity analyses that include correlated variables (except for three-way sensitivity analysis) will display a text label including the name and the range of each correlated variable which was varied during the analysis. On the graph, this text will be placed adjacent to the name of the original variable at the appropriate axis.

Additional sensitivity analysis topics

Other chapters cover additional software features which can be useful in testing the sensitivity of your trees, but whose use is not restricted to performing sensitivity analysis. See, for example:

- Creating analysis sequences, in Chapter 19
- Using variable sliders, in Chapter 13
- Linking trees, in Chapter 21
- Probabilistic sensitivity analysis, in Chapter 16

Sensitivity analysis on variables with non-numeric definitions

A sensitivity analysis can be performed on any variable in your tree, whether it has a numeric value definition (e.g., X=1 or X=Exp(2)) or a variable expression (e.g., X=Rate*Util). When performing a sensitivity analysis on a variable defined as a formula, you have multiple options. You can perform a sensitivity analysis on the component variables (e.g., Rate and Util) using variable correlations or a multi-way sensitivity analysis. Alternatively, you can perform a one-way sensitivity analysis on the original variable (e.g., X) based on an estimated numeric value range.

If you treat X as the independent variable, however, the formula will be ignored during the course of the analysis. Definitions of Rate and/or Util at different points in your tree will not be used during this analysis.

It is advisable to focus sensitivity analysis on the finest-grain parameters. In the example above, X is no longer "finest-grain," as it has been defined in terms of its two component variables. In general, models should be designed to ensure that the sensitive variables have a single numeric definition.

Checking probability coherence

Most forms of sensitivity analysis offer an option labeled Check coherence. When this option is selected, TreeAge Pro will ensure that, at each interval, (i) all probabilities sum to 1.0 and (ii) no probabilities are negative. The analysis will be halted if at any time either rule is violated.

If the subject variable is used to define a probability, you are encouraged to leave this option selected. This will ensure the validity of your model over the range of the analysis. This is particularly important in the initial stages of testing your model's validity. The downside is that calculation time is increased. If calculation speed is a concern, and you are not including any probability variables in the sensitivity analysis, you may want to turn off coherence checking.

Also see the section on probability non-coherence in Chapter 22.

Analyzing a single option

Normally, when performing a sensitivity analysis, a decision node is selected and TreeAge Pro displays one line for each of the alternative scenarios rooted at the selected node. It is possible to focus a one-way sensitivity analysis on a single scenario, rather than on all of the scenarios emanating from a decision node.

If the node you select prior to performing the sensitivity analysis is not a decision node, TreeAge Pro will assume that the results should be presented as a single line. This will represent the changing expected value of the scenario rooted at the selected node. (Healthcare module users: Note that this option is not available for cost-effectiveness sensitivity analyses, which must be performed at a decision node.)

If, however, you select a decision node which is an immediate descendant of a decision node, TreeAge Pro will give you the option of drawing one line for the selected node (as a branch of its parent), or multiple lines for the branches emanating from the selected decision node.

Chapter 16

Using Distributions and Monte Carlo Simulation

This chapter provides instructions on creating and referencing sampling distributions, performing Monte Carlo simulation, and interpreting the results.

Chapter 15 describes the properties of some of the commonly-used built-in distributions in TreeAge Pro. Instructions on creating custom, table-based distributions can be found at the end of Chapter 17.

For detailed information on using the Healthcare module to perform probabilistic sensitivity analysis and microsimulation on Markov and cost-effectiveness models, see Chapters 33 and 36.

In this chapter:

- Uses of Monte Carlo simulation 202
- Creating distributions 205
- Performing probabilistic sensitivity analysis 209
- Monte Carlo simulation reports and graphs 211
- Simulation options 214
- Two-dimensional simulation options 217

Uses of Monte Carlo simulation

Each of the analyses described in previous chapters, including sensitivity analysis, is *deterministic*. There is no randomness in these types of model calculations; during each calculation, each model parameter uses its specified point value. If an analysis is repeated using the same parameters, the results will be unchanged.

In contrast, there are many situations where introducing a random, or *stochastic*, element into some part of the analysis can be useful. In these situations, Monte Carlo simulation can be applied.

Probabilistic sensitivity analysis (sampling)

Monte Carlo simulation refers to the use of random numbers in evaluating a model. Perhaps the most frequent application of Monte Carlo simulation in TreeAge Pro is as a form of sensitivity analysis. Like regular *(deterministic)* sensitivity analysis, Monte Carlo simulation also recalculates a model multiple times. Monte Carlo simulation can update any number of parameters between model recalculations, assigning values that are randomly sampled from probability distributions. This use of Monte Carlo simulation is referred to as *probabilistic* sensitivity analysis.

One advantage of probabilistic sensitivity analysis is that all parameter uncertainties can be incorporated into an analysis (see section on nonlinearity, below). Sampling parameter values from probability distributions (rather than from a simple range defined by upper and lower bounds) places greater weight on likely combinations of parameter values, and simulation results quantify the total impact of uncertainty on the model, in terms of the confidence that can be placed in the analysis results.

This chapter focuses on the basics of performing probabilistic sensitivity analysis in TreeAge Pro.

Microsimulation v. PSA:

Simple microsimulation and probabilistic sensitivity analysis (PSA) have separate applications and different interpretation. Note that commonly-used PSA outputs (e.g., value of information curve, acceptability curve, ICE scatterplot) may not have an intuitive application in a microsimulation analysis. (In other words, the distribution of microsimulation outcomes is not interpreted the same way as the distribution of outputs from a PSA.)

Microsimulation (first-order trials)

In TreeAge Pro, Monte Carlo simulation includes two distinct features, which may be used separately or in combination: sampling parameter values, described above (sometimes called second-order simulation); and running random trials (also called first-order simulation, microsimulation, or a random walk). These two kinds of simulation correspond roughly to two categories of uncertainty — second-order, *parameter uncertainty* versus first-order uncertainty (*variability* among individuals, or over time) — and have different applications and methods. First-order trials are described conceptually here; their use in Markov Monte Carlo simulation in the Healthcare Module is covered in Chapter 36.

The last step in each iteration of a second-order simulation is recalculating the model. The most efficient way to perform this recalculation is using expected value calculations, but it is also possible to use first-order simulation as a means of approximating an expected value.

First-order simulation trials can be used to model the variability in individual outcomes, visualized in a decision tree as the branches of a chance node. Simulation trials use random numbers to select a single path through the tree, following one branch at each chance node, with higher probability events being more likely. Running 100 first-order simulation trials results in a list of 100 individual outcomes (e.g., profit equals $150, $175, $0, $550, -$50, and so on), with some chance of repeating outcomes. As more individual trials are run through a decision tree, the average outcome should approach the regular expected value calculation. (Increasing the numbers of trials will also result in a standard deviation for the simulation that should converge on the expected value form of standard deviation; see Chapter 5.)

First-order trials have somewhat limited use with most models (with the major exception of Markov models). One possible use of simulation trials in a regular tree is to replace any chance node with a parameterized probability distribution (e.g., sampling an outcome from a normal distribution, or any other continuous or discrete sampling distribution).

For example, in the investment decision tree, the risky investment's chance node could be replaced with a distribution representing either a continuous range of outcomes, or a discrete distribution just like the existing three-branch chance node. This can also be accomplished without simulation trials, using TreeAge Pro's DistKids( ) function; these advanced uses of distributions in modeling are covered in Chapter 22.

Simulation using multiple processors

To support the complex, potentially lengthy types of analysis described in this chapter, TreeAge Pro can utilize up to eight processors on a single computer when performing Monte Carlo simulation. Or, with the Distributed Simulation module, TreeAge Pro can divide the simulation among helper computers. See Appendix F for additional technical details on multi-processor calculations.

Markov Monte Carlo microsimulation

For information on using 1st-order simulation trials with Markov models and tracker variables, refer to the TreeAge Pro Healthcare module documentation in Part V of the manual, including Chapter 36.

Nonlinearity

In some models, calculating an expected value based on the *mean* value of an uncertain parameter (i.e., using roll back) is equivalent to randomly sampling many values for the uncertain parameter from its probability distribution, recalculating the model for each sample, and taking the average. However, this is not the case with all models and all parameters.

If, for example, a distribution is used to define an uncertain component of a probability or utility function, you may find that the "expected value" of the model calculated using the parameter's mean value will differ from the average of many recalculations of the model using sampled values for the uncertain parameter. In these cases, the simulation average value is the better "expected value" for the model (and therefore simulation would be the preferred means of analyzing the model).

EVPI/value of information analysis

Monte Carlo simulation can be used to perform various kinds of "value of information" analysis, similar to the structural form of EVPI described in Chapter 5.

The Analysis > Expected Value of Perfect Information command in TreeAge Pro calculates the difference between the baseline expected value of a decision, and the expected value when a chance node is temporarily shifted to the left of the decision. In a Monte Carlo simulation, the calculation of EVPI is done differently.

For example, if the optimal strategy changes for different sampled parameter values, then there is some benefit to having "perfect information" about the uncertainty prior to the decision. The average of the values of the best option from each recalculation is the expected value *with* perfect information; it will either be equal to or greater than the best average value for any single alternative. Calculating the difference gives the expected value *of* perfect information.

TreeAge Pro v2005 added an EVPI report and chart to the Monte Carlo simulation window's **Graph** popup menu. These are described later in this chapter.

TreeAge Pro also includes an option to do two levels (nested loops) of parameter sampling during simulation, in order to do robust EVPI-type simulation. This may be required to get an accurate EVPI when there is nonlinearity in the model, as described above. The "information" parameter is sampled in the top-level, outside loop; for each sample value, the model is reevaluated using an inner loop (a simulation of N iterations which samples the remaining uncertainties).

Creating distributions

As detailed in the previous chapters on performing deterministic, *n*-way sensitivity analysis in TreeAge Pro, parameters targeted for analysis must be defined using **variables**. Similarly, before using Monte Carlo simulation to perform probabilistic sensitivity analysis in TreeAge Pro, uncertain parameters must be defined using **distributions**.

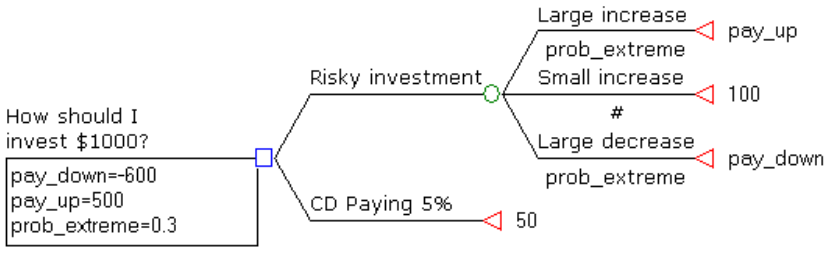
Probability distributions can be employed in any kind of formula in a tree, including variable definitions, payoffs, rewards, probabilities, and even in the parameters of other distributions. TreeAge Pro includes more than 20 different built-in distributions, and can also sample from empirical data in tables.

In TreeAge Pro, distributions that you define are stored in a list in the tree and assigned an integer index and, optionally, a name (like a variable name). A distribution can be used in a tree calculation either by referencing it by name, or by using the Dist(*n*) functions, where n is the distribution's integer index.

This chapter describes the basic steps required for **probabilistic sensitivity analysis**:

- Define distributions in your tree;
- Use the distributions in calculations;
- Sample from distributions and recalculate the model using a Monte Carlo simulation.

To illustrate, the simple investment model will be used in the tutorial. The tree with variables, but no distributions yet, is shown below.



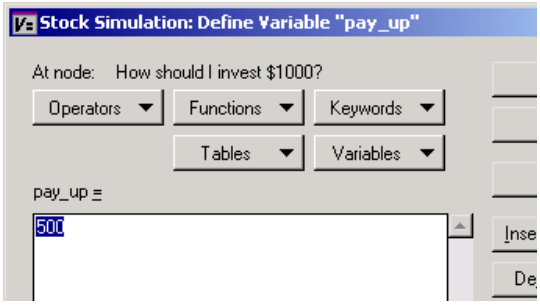
Defining a parameter using a distribution

The first steps are to change the payoff of the Large increase terminal node from a point estimate of 500 to a Normal distribution with a mean of 500 and a standard deviation of 100.

► **To define a distribution in the tree:**

▷ Open the Three Variables version of the tree that you built in the sensitivity analysis tutorial in Chapter 12. (A copy of this file, called Three Vars, is in TreeAge Pro's Tutorial Examples subdirectory.) Save a copy, calling it Stock Simulation.

▷ A variable pay_up is already defined at the root node = 500. Right-click on the root node, and choose **Define Variable >** pay_up.

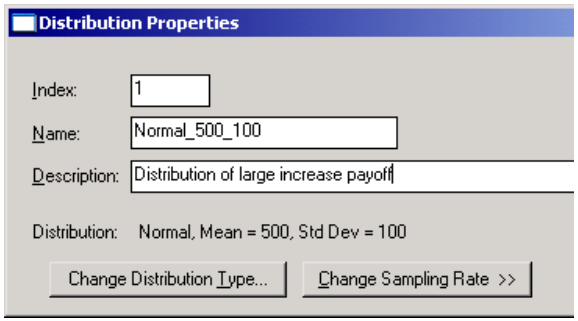


▷ In the Define Variable window for pay_up, delete the current value, and click on the button labeled **Insert Distribution….**

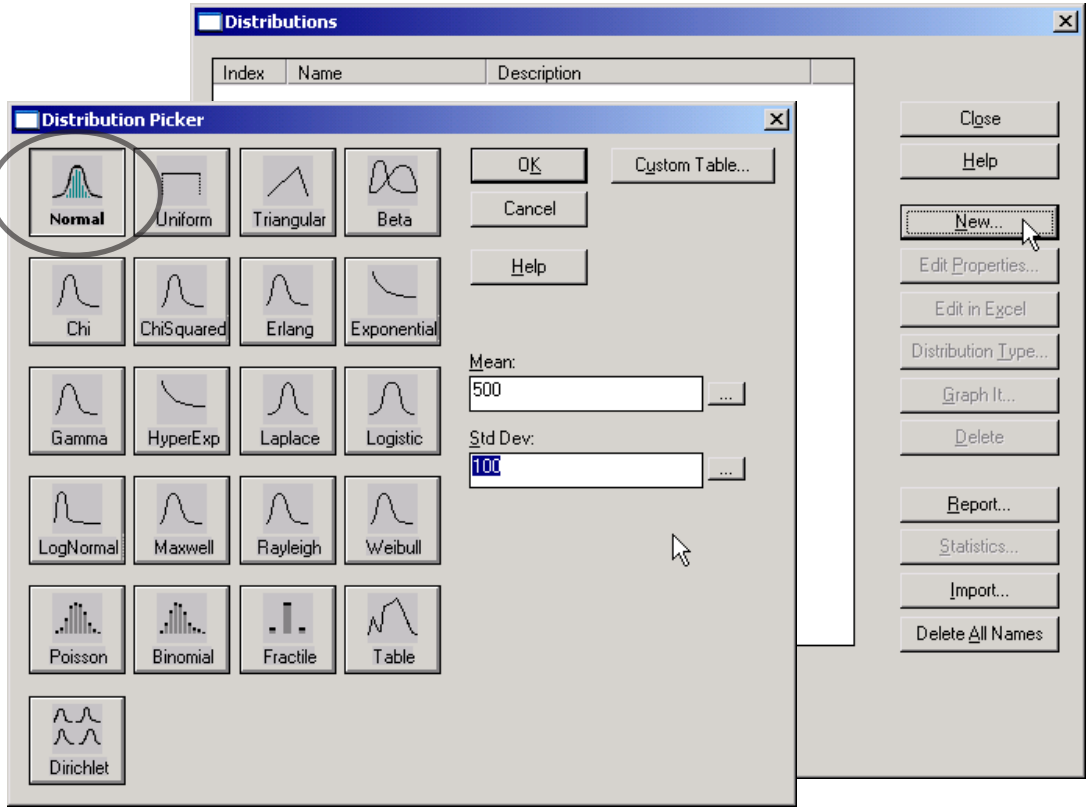


The Distributions list for the tree will appear. No distributions are listed because none have been defined in the tree yet.

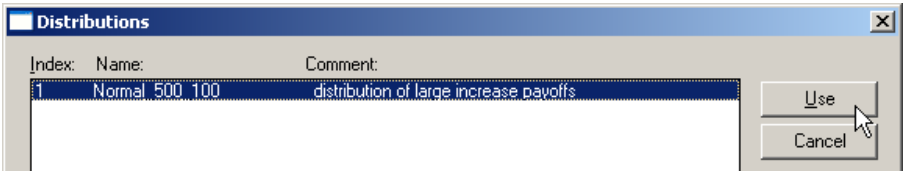
- ▷ To add a distribution to the list, click **New…**.
- ▷ From the Distribution Picker palette of distributions, shown below, click on the button corresponding to the desired distribution type. In this case, use a Normal distribution for the payoff.
- ▷ Enter the required parameters for the selected distribution. Parameters can be numeric, or variables. In this case, define a mean of 500 and a standard deviation of 100 for the Normal distribution. Press ENTER or click OK to save the distribution and its parameters.
- ▷ The Distribution Properties dialog appears, showing the distribution's integer **index** (automatically assigned).



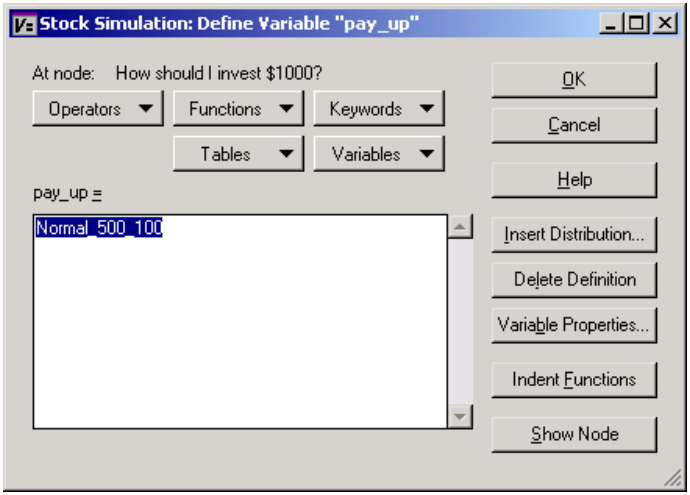
- ▷ An optional name and descriptive comment can be entered here. Enter `Normal_500_100` for the name, and `Distribution of large increase payoff` for a description.



▷ Click OK to save the distribution and return to the Distributions list, which should show the new Normal distribution.

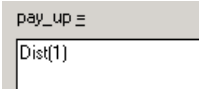


▷ With the new distribution selected in the list, click the **Use** button. TreeAge Pro will insert a reference to the distribution in the active expression editor.



If you assigned a name to the distribution, this will be inserted into the variable definition.

If you do not name the distribution, TreeAge Pro will instead insert a reference like "Dist(1)" with the integer index of the distribution entered inside the parentheses.



▷ Click OK to close the Define Variable window.

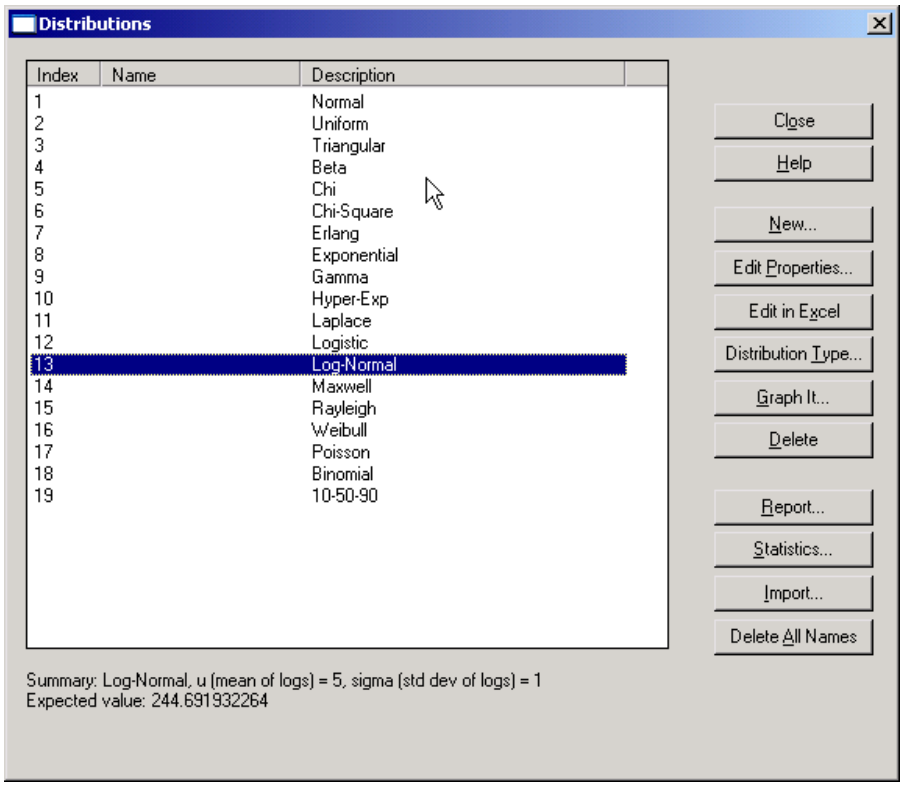
Distribution notes:

- During roll back and other non-sampling analyses, a distribution is equal to its mean value. In the example, pay_up will still equal 500 during expected value calculations, because this was specified as the mean value of the Normal distribution.
- The Dist() function in TreeAge Pro is equivalent to the DistSamp() function in DATA and earlier versions. Other versions of the distribution function, including DistForce( ) and DistKids( ), are described in Appendix D.
- During a single iteration of a sampling simulation, all references to a particular distribution return the same sample value (with the exception of any distribution set to resample more frequently, for example in Markov microsimulations).
- Each named distribution is added to a list at the root node of the tree, which is normally hidden and not directly editable. The list is visible both in the tree's Properties grid and the Variable Definitions window. To show the list on the face of the tree, turn on the display of global definitions in the Variables\ Markov Display preferences.

Managing distributions

The Distributions dialog can also be opened from the tree window, by choosing **Values >** Distributions…. Use the Distributions dialog to add, delete, edit, export, test, and report on distributions.

- It is not necessary to delete a distribution in order to change its parameter values or distribution type. Simply select it in the list and click the **Distribution Type…** button. To change its name, comment, or other properties, click **Edit Properties…** instead.
- If your license includes the Excel module, the **Edit in Excel** button can be used to export selected distributions to a table in a new Excel worksheet, where they can be edited. From Excel, the add-in's **TreeAge >** Add or Update Distributions command can be used to update the active tree with any changes. See Chapter 28 for details.
- The **Import…** button can be used to remove the existing list of distributions from the tree, and replace it with a list from a different, saved tree.
- To preview the shape or statistics of distributions before simulating the entire model, use the **Graph It...** or **Statistics** buttons.
- The different types of continuous and discrete distributions that are available in TreeAge Pro are described in more detail in Chapter 16.

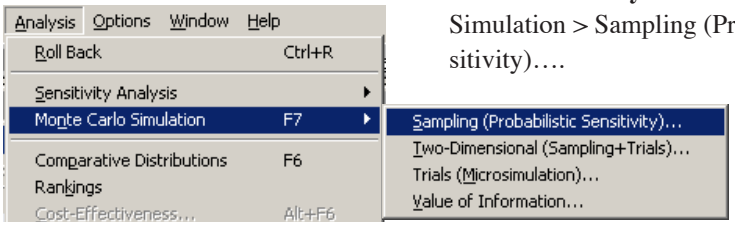


Performing probabilistic sensitivity analysis

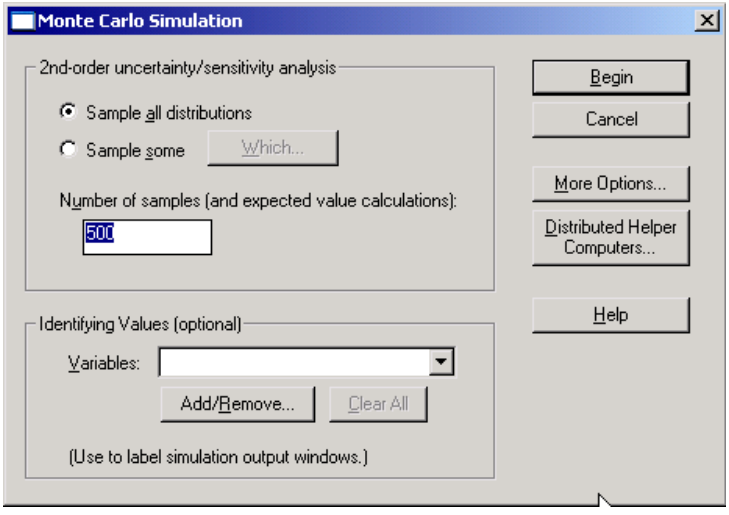
After defining and using distributions in your tree, you can run a Monte Carlo simulation to see how resampling parameters values affects calculations at a selected node. Simulations can be run at any node except a terminal node, making it possible to analyze only part of a tree.

► **To perform a probabilistic sensitivity analysis/Monte Carlo simulation:**

▷ In a tree which uses sampling distributions, select a decision node, and choose **Analysis >** Monte Carlo Simulation > Sampling (Probabilistic Sensitivity)….



If you have defined distributions in the Distributions dialog (even if they are not *used* anywhere in the tree), the Monte Carlo Simulation dialog will present a variety of options for the analysis. In this case, you simply specify the desired number of distribution samples and corresponding model recalculations.



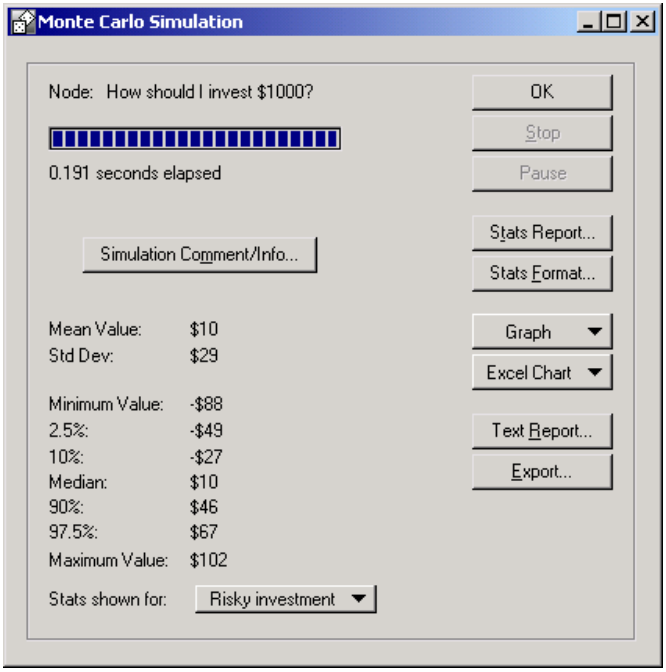
Now, specify the settings to use for this simulation.

- ▷ Change the number of distribution samples (and recalculations of the entire model) to `500`, and leave other settings at their defaults (to sample all distributions).
- ▷ Click Begin to start the simulation.

In running this simulation, TreeAge Pro will randomly sample 500 values from the distribution defined in the tree, and recalculate expected values for the branches of the selected decision node based on each parameter sample.

The simulation output window shows the progress of the simulation. During long running simulations, partial results can be reported. Click the **Calc Stats** button to display a running statistical summary of the completed iterations. It is also possible to end a simulation before it is completed, using the **Stop** button, and still review the partial results.

Once the simulation is complete, a final statistical summary will be displayed. Unless you use "seeding," there should be statistical variation from one simulation to the next.

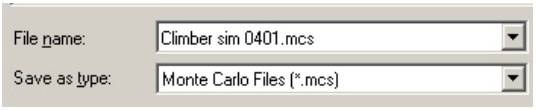


Saving Monte Carlo simulations

Depending on the complexity of the model, and the number of samples and recalculations you specify, running a simulation can be time consuming. For this reason, the simulation output window can be saved, separately from the tree, as a Monte Carlo Simulation (*.MCS) file. Saving the simulation output will allow you to share the complete results with other TreeAge Pro users, or to generate graphs and reports from the simulation at a later time.

► **To save the Monte Carlo output window:**

- ▷ Choose **File >** Save (or Save As…).

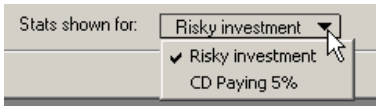


Simply use the **File >** Open command to reopen a Monte Carlo simulation (*.mcs) file.

(***NOTE:*** MCS files grow relative to the number of rows and columns. Sizes of ~1 GB would be achieved with 1 million rows and 150 output columns. Use a Zip compression program to shrink file size.)

Simulation reports and graphs: the basics

If a simulation is performed at a decision node, the statistics for the top branch are displayed by default. Statistics for other branches of the decision node are accessible using the **Stats shown for** menu button, at the bottom of the window.

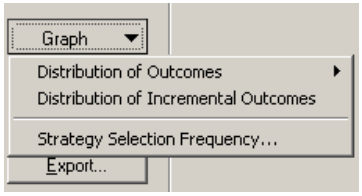


To display statistics (for both inputs and outputs) in a regular text report window, click the **Stats Report…** button. The information can be copied to the clipboard or saved to a text file.

Clicking the **Text Report** button will display the list of iterations, distribution samples, and corresponding tree recalculations and other outputs.

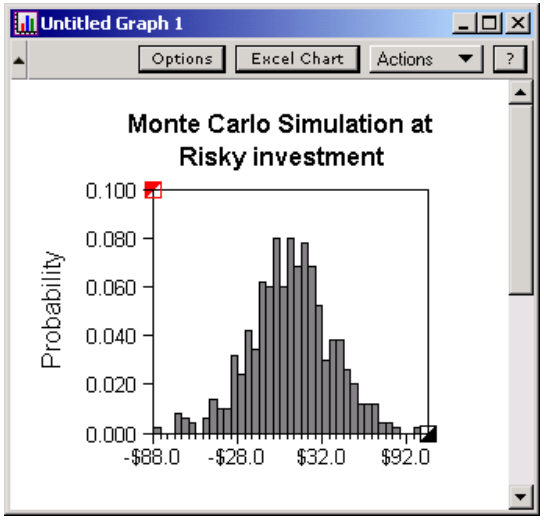
The text report window can handle thousands of rows, but you may prefer to export the results to a text file using the **Export** button, which will output the simulation results directly to a file.

Clicking on the **Graph** button lets you choose from among a variety of TreeAge Pro's built-in graph types.

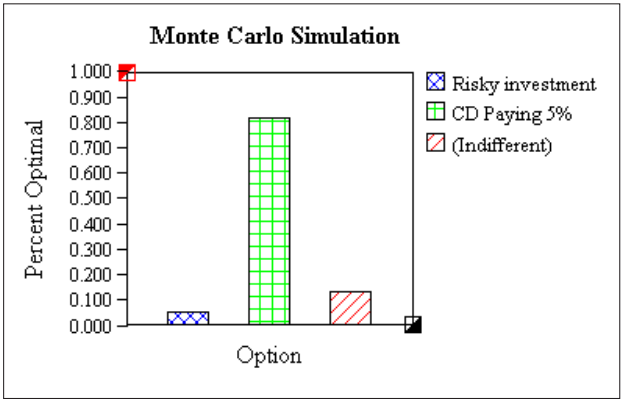


For all simulations, probability distribution histograms can be displayed for the model calculations, as well as for any sampling distributions.

For a discussion of probability distribution histograms, and the options available in TreeAge Pro for working with and customizing distribution graphs, see Chapters 5 and 6.

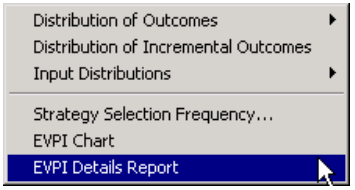


For a simulation at a decision node, it is possible to view the percentage of samples for which each alternative is optimal, using the **Strategy Selection frequency** graph. An indifference tolerance value can be specified.



Expected Value of Perfect Information (EVPI)

Probabilistic sensitivity analysis simulations performed at decision nodes include a detailed EVPI report and summary chart. Both the report and chart are included under the **Graph** popup menu.

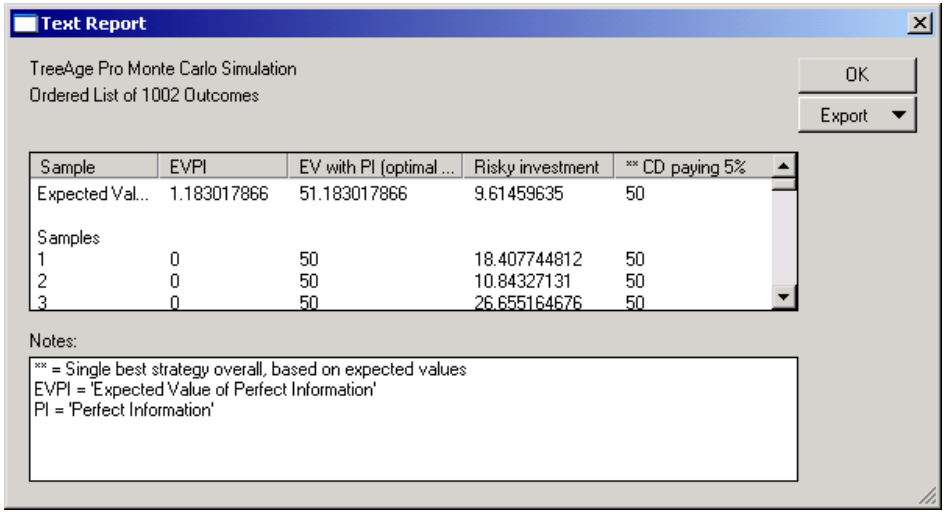


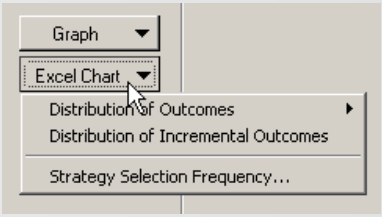
Note:
The EVPI reported for a *microsimulation* (versus a sampling simulation) might be interpreted as the value of being able to predict all chance nodes' outcomes. Using the Stock Tree, for example, compare the Analysis > Expected Values of Perfect Information result at the single chance node in the model to the simulation EVPI report/chart in a simulation performed at the decision node.

The calculation of EVPI in simulations is relatively straightforward:

- determine the **overall optimal strategy**, using the simulation mean values (for cost-effectiveness models, uses net monetary benefits based on a specified threshold ICER);
- determine the **optimal strategy for each sample iteration** in the simulation (normally the outermost loop will be a sampling loop, although TreeAge Pro will do similar calculations even if it is just a microsimulation with no sampling);
- for each iteration, if the optimal strategy is not the overall optimal strategy, calculate its incremental value (iteration optimal - overall optimal), which will be >= 0;
- report the average (expected) value of perfect information over all iterations.

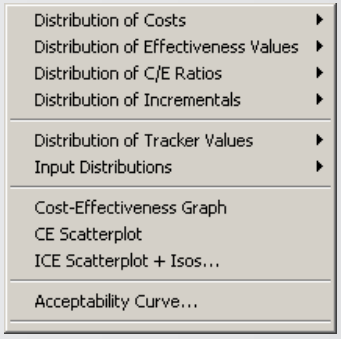
In addition to the full EVPI text report, a simple chart shows the EVPI alongside to the EV of the single best strategy overall.


Text Report
TreeAge Pro Monte Carlo Simulation
Ordered List of 1002 Outcomes
OK
Export
Sample | EVPI | EV with PI (optimal ... | Risky investment | ** CD paying 5%
Expected Val... | 1.183017866 | 51.183017866 | 9.61459635 | 50
Samples
1 | 0 | 50 | 18.407744812 | 50
2 | 0 | 50 | 10.84327131 | 50
3 | 0 | 50 | 26.655164676 | 50
Notes:
** = Single best strategy overall, based on expected values
EVPI = 'Expected Value of Perfect Information'
PI = 'Perfect Information'



Charting/Reporting with Microsoft® Excel

If you have the TreeAge Pro add-in for Excel, graphs generated as TreeAge graph files can also be created as Excel charts. In some cases, Excel charts have different/additional functionality. See Chapter 28 for details.



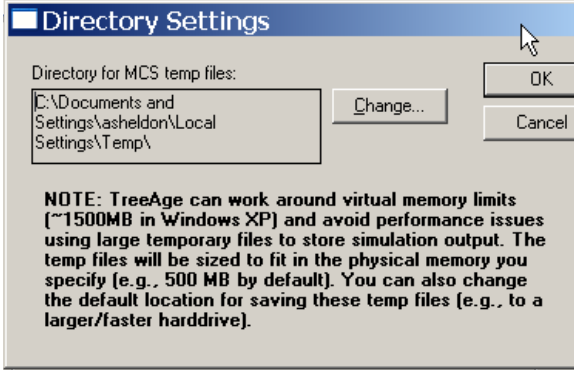
Cost-Effectiveness Simulation Outputs

Numerous additional graphs and reports are available in cost-effectiveness simulations performed using the Healthcare module for TreeAge Pro. For example, in a cost-effectiveness simulation, the Acceptability Curve in used in place of the Strategy Selection graph. See Chapter 33 for additional details.

Simulation options

There are a wide variety of types of customizations that can be made to simulations in TreeAge Pro.

> **Tip:**
>
> Check the Save settings box, to keep these setting for subsequent simulations of this tree.


Directory Settings
Directory for MCS temp files:
C:\Documents and Settings\asheldon\Local Settings\Temp\
Change...
OK
Cancel
NOTE: TreeAge can work around virtual memory limits (~1500MB in Windows XP) and avoid performance issues using large temporary files to store simulation output. The temp files will be sized to fit in the physical memory you specify (e.g., 500 MB by default). You can also change the default location for saving these temp files (e.g., to a larger/faster harddrive).

Distributed simulations

With the Distributed Simulation module running on helper computers, TreeAge Pro can divide the simulation among multiple computers for faster simulation. For more details, go to:

www.treeage.com/support/distributedmc.html

Multi-threading for faster simulations

To better handle lengthy simulations, TreeAge Pro will use up to eight processors on a single computer when performing Monte Carlo simulation.

If you are running simulations on a dual-processor computer, you can instead specify that TreeAge Pro use only one simulation thread, which will leave one processor idle during the simulation (allowing other programs to run quickly while the simulation is running). If you are running a multi-threaded simulation, it is recommended that you leave the Optimize all expressions… setting on.

Conserving memory

Virtual memory requirements of very long simulations can be very high. Under Windows XP for example, reporting 1 million rows for a tree with 120 output columns (e.g., C/E for 5 strategies, with 22 trackers) would previously exceed virtual memory allowed a program (~1.5GB).

To avoid this problem, the user can specify a threshold memory limit (default=500MB) above which TreeAge will swap memory to temporary files.



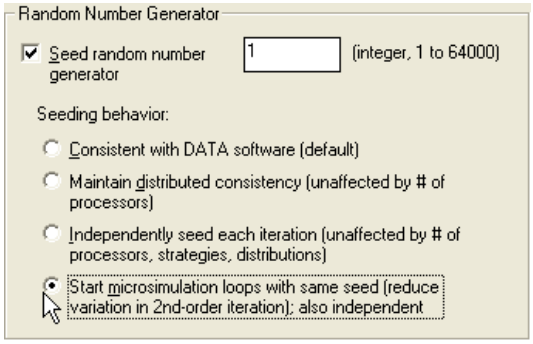
"Seeding" the random number generator

Normally, Monte Carlo simulation uses dynamic information from the computer's clock to initialize a sequence of pseudo-random numbers. If you ever need to force the same set of samples and/or trials for several different simulations of the same model, you can specify that TreeAge Pro use the same, predictable sequence of pseudo-random numbers.

► **To seed a simulation:**

▷ In the setup dialog for a Monte Carlo simulation, click the **More Options…** button.

▷ Check the Seed random number generator box, and specify an integer seed value (from 1 to 64,000).


Random Number Generator
Seed random number generator
1
(integer, 1 to 64000)
Seeding behavior:
Consistent with DATA software (default)
Maintain distributed consistency (unaffected by # of processors)
Independently seed each iteration (unaffected by # of processors, strategies, distributions)
Start microsimulation loops with same seed (reduce variation in 2nd-order iteration); also independent

The underlying behavior/algorithm used during seeding can be changed, if necessary:

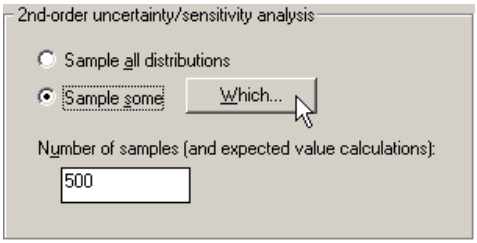
- **"Distributed consistency"** ensures that the sequence of random numbers does not change depending on the number of processors used.
- "**Independently seed each iteration**" generates a sequence that also does not change if strategies are reordered or removed, for example.
- "**Start microsimulation loops…**" is used for variance reduction between repetitions of microsimulation loops within two-dimensional simulations.

Turning off sampling of selected distributions

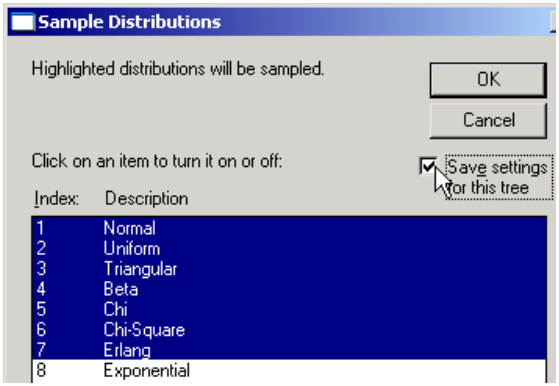
You have the option of specifying that only a subset of the distributions should be sampled during a simulation. Non-sampled distributions will be set at their mean.

► **To sample from selected distributions during a simulation:**

▷ In the simulation setup dialog, select the **Sample some** option, and click the **Which...** button.



▷ In the Sample Distributions dialog, selected distributions will be sampled. To turn on or off the sampling of a particular distribution, click on it in the list.



▷ Click OK to return to the Monte Carlo Simulation dialog.

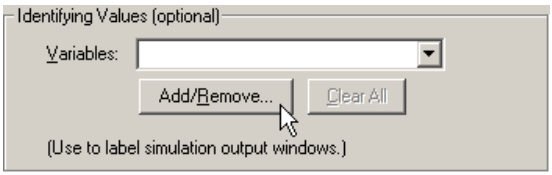
Identifying simulations

Simulation output windows can be labeled using the values of variables in your tree. This is useful when you have more than one simulation output window open at once, or if you are saving and reopening or sharing the Monte Carlo simulation files. The resulting simulation output window will display a list of the variables and their default values.

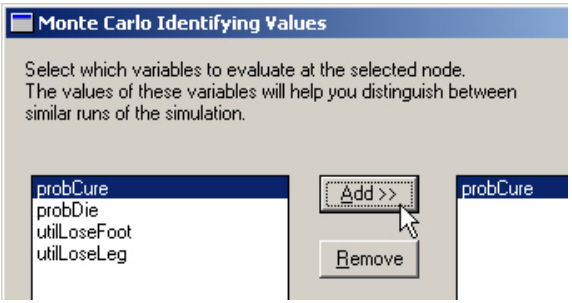
Monte Carlo simulation identifying values are particularly useful in combination with the sequenced analysis feature in TreeAge Pro; see Chapter 19.

► **To add identifying variable values in the simulation window:**

▷ Click the **Add/Remove...** button in the Identifying Values section at the bottom of the Monte Carlo Simulation dialog (see the next page).



▷ In the Monte Carlo Identifying Values dialog, select tree variables from the list on the left, one at a time, and click the Add button to include them in the identifying values list.



▷ Click OK to return to the Monte Carlo Simulation dialog.

The identifying values list will be displayed in all subsequent simulation windows generated for that tree. To clear the entire list, click the **Clear** button in the Monte Carlo Simulation dialog's Identifying Values section. To remove only selected variables from the identifying values list, use the **Remove** button inside the Monte Carlo Indentifying Values dialog.

Simulation comment

Simulation output windows can also be labeled using the Comments button. Use this field for narrative description of the simulation. It is pre-populated with an abstract of the simulation.

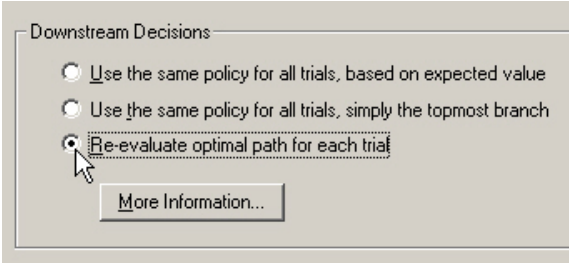
"Downstream" decision nodes during simulations

If your tree includes decision nodes to the right of the node where a simulation is being performed, each recalculation of the model must select a single path when it encounters such *embedded* decisions.

These embedded decisions can be handled in any of three ways: 1) based on expected values calculated prior to the simulation, a single optimal policy can be followed for all iterations; 2) if any distributions are defined, the optimal policy can be reevaluated for each iteration, based on the current distribution sample values; or 3) the topmost branch can always be selected as optimal.

► **To set the downstream decision algorithm for a simulation:**

▷ In the Monte Carlo Simulation setup dialog, click on the **More Options…** button.


Downstream Decisions
Use the same policy for all trials, based on expected value
Use the same policy for all trials, simply the topmost branch
Re-evaluate optimal path for each trial
More Information...

Two-dimensional simulation options

In most models, a probabilistic sensitivity analysis should recalculate **expected values** for each new set of parameter samples, to see the effects of parameter uncertainty.

However, there is an alternative method for recalculating the tree after each set of parameter samples: estimating an expected value by averaging many **first-order trials**. This process – estimating expected values by averaging a sufficiently large number of random, individual outcomes – is sometimes called microsimulation.

Recalculate using first-order trials (microsimulation)

In most models, two-dimensional simulation for the purposes of probabilistic sensitivity analysis is unnecessary. It is normally preferable to run a 1-dimensional loop that recalculates EVs for each set of randomly sampled parameter values – except in the case of certain types of models, for example:

- Markov models using tracker variables to follow detailed event history. See Chapter 36 for details on trackers.
- Trees using distributions to represent variability among individuals (much like a chance node does) rather than just parameters whose values are uncertain. See Chapter 17 for more information on sampling rates.

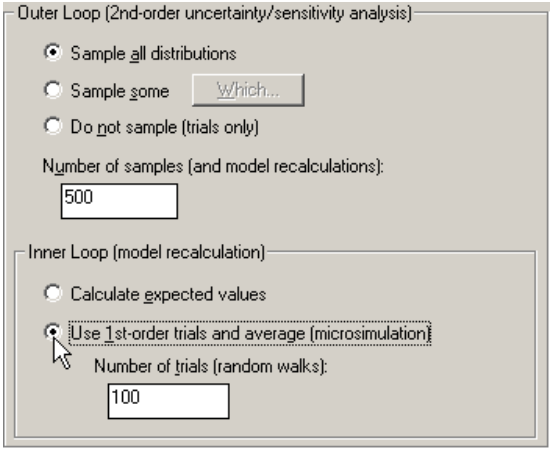
Both of these situations call for microsimulation (described at the beginning of this chapter).

Depending on the complexity of the model, ***each microsimulation*** may require thousands of first-order trials in order to adequately approximate an expected value. In the context of a probabilistic sensitivity analysis, therefore, using a microsimulation for each recalculation of the tree (for a new set of samples) can be very time consuming. This is why two-dimensional simulation is generally avoided unless a model absolutely requires microsimulation to calculate correctly.

► **To perform probabilistic sensitivity analysis using first-order microsimulation trials for model recalculations:**

▷ Select the root node of your tree, or a node to the right, to set the context for analysis.

▷ Choose **Analysis >** Monte Carlo Simulation > Two-Dimensional…. .



▷ If your model requires microsimulation for the EV step, change the "Inner Loop" setting to Use 1st-order trials….

▷ Specify a sufficient number of 1st-order trials to get a good average for each new sample value (or set of values). Initially you should experiment with smaller number of iterations in both the sampling loop as well as the inner, microsimulation loop.

Refer to Chapter 36 for more information on two-dimensional simulations, in the context of Markov models using trackers with microsimulation.

Two-dimensional simulation details

In models requiring microsimulation, however, a two-dimensional (or two-loop) simulation approach is used:

1. *N* number of parameter samples and *I* number of microsimulation trials are specified
2. Draw one set of samples for the parameter distributions.
3. Holding the sampled parameter values constant, a group of *I* microsimulation trials are performed. The mean of the results from the trials is calculated (for both calculated cost and/or effectiveness values, as well as for tracker variables).
4. Steps 2 and 3 is repeated for each of the *N* sets of parameter samples.

Once the simulation is complete, the final report will include *N* rows, showing each set of parameter samples and the corresponding mean values for *I* microsimulations. The distribution of these results reflects the total uncertainty resulting from the parameter distributions.

Three-dimensional simulations

First-order trials can also be used in the evaluation step of a Value of Information simulation, resulting in a three-dimensional simulation.

Chapter 17

Distribution Functions, Options, and Types

This appendix provides infromation about each of the built-in sampling distribution types available in TreeAge Pro.

Derivations and detailed explanations of the distribution formulae provided here may be found on many math/statistics web sites, and in most texts on probability theory. See, for example, Christensen, Ronald; ***Data Distributions: A Statistical Handbook*** (2nd Ed.); Lincoln, Massachusetts: Entropy Limited, 1989.

See Chapters 15, 16, and 22 for information on using distributions in trees.

In this appendix:

- Distribution functions 220
- Distribution options 221
- Commonly-used distributions 224
- More continuous distributions 225
- More discrete distributions 229
- Sampling from tables during Monte Carlo simulation 230

Distribution functions

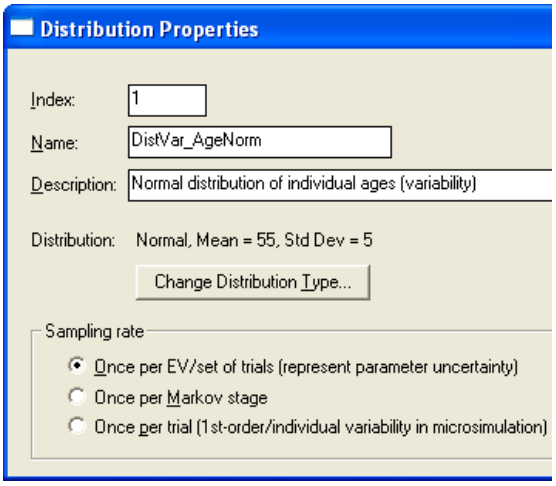
Distributions can be referenced in two different ways: by name (if an optional name is defined in the distribution properties) or by index. The following functions employ the argument *index* to reference a distribution defined for the tree (see Chapter 16). With the exception of DistKids(), they are intended primarily for use during Monte Carlo simulation.

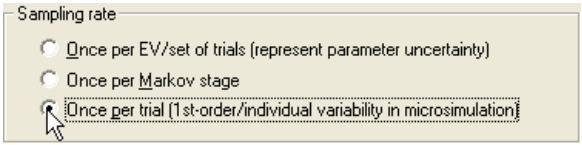
Dist(index)	During expected value calculations, this function returns the indexed distribution's mean. During Monte Carlo simulations, this function returns the most current random sample for the distribution (unless the distribution is flagged to not sample).
Dist(index; column)	For a multi-variate distribution (i.e., a Dirichlet or multi-column table distribution), returns the sample value (or mean, if not sampling) from the designated value column.
Dist(index; 1)	For a univariate distribution (i.e., not Dirichlet or multi-column table), forces a sample from the distribution, even during non-sampling analyses (see DistForce below).
Dist(index; 2)	For a univariate distribution (i.e., not Dirichlet or multi-column table), returns the stored sample value for a virtual branch created using the DistKids( ) syntax, below.
DistForce(index; …)	Unlike the Dist(*index*) syntax, DistForce() always samples a value, even during non-sampling analyses such as roll back. DistForce(index; n) also works for multi-variate distributions. Note that DistForce(index; 1) must be called first, in order to trigger a new set of sample values for all columns. Then the stored sample values for the other columns can be referenced.
DistKids(index; samples)	A probaiblity function which dynamically creates invisible branches (number = samples) at a chance node during any calculation. Used in combination with the Dist(index; 2) syntax described above.
DistSamp(index)	Obsolete. See Dist().
DistProb(index; val)	For the specified distribution, returns the approximate cumulative probability of the specified value.
DistTrim(index; min; max)	Same as the Dist() function, except it will resample (up to a **maximum of 10 times**) until a sample is returned that falls between the specified minimum and maximum values.
DistValue(index; prob)	For the specified distribution, returns the approximate value at the specified cumulative probability.
Seed(*n*)	Sets the random number generator in the current thread (multi-processor simulations use multiple threads). Use a positive integer argument. If no argument is given, or a zero argument, a pseudo-random seed is generated (i.e., as if no seeding).

Do you have suggestions for new distribution functions/syntax? Send an e-mail to support@treeage.com.

Distribution options

Some special options can be used to make distributions more flexible.




Sampling rate
Once per EV/set of trials (represent parameter uncertainty)
Once per Markov stage
Once per trial (1st-order/individual variability in microsimulation)

Microsimulation and PSA:

Simple microsimulation and probabilistic sensitivity analysis (PSA) have separate applications and different interpretations. However, PSA can be run on models requiring microsimulation, in a two-dimensional simulation. If you are planning to run two-dimensional simulations, understanding the sampling rate options for distributions is important (in case you have any distributions that represent variability rather than uncertainty).

Changing sampling rate for microsimulation models

While most distributions are used to ***sample possible values of uncertain parameters*** for probabilistic sensitivity analysis, distributions can also be used to instead represent ***individual variability/patient characteristics***. These two different classes of distributions should be identified via the TreeAge Pro distribution's sampling rate property. This is particularly important if a model includes ***both*** types of distributions.

► **To change the sampling frequency for a particular distribution:**

▷ Open the Properties dialog for the distribution by double-clicking on the distribution in the Distribution list for the tree. (Excel module users can edit this property in Excel; see Chapter 28.)

▷ To identify the distribution as representing per-trial variability, click on the bottom option, **Once per 1st-order trial**.

This option will be of interest in specific models:

- Markov models using tracker variables to follow detailed event history, including discrete event models.
- Trees using some distributions to represent variability among individuals (in the same way that a chance node could).

By default, distributions are set to sample once per tree EV recalculation. In a two-dimensional simulation, this would be per "group of trials". By default, a simple microsimulation resamples only distributions set to sample per 1st-order trial or Markov stage, unless you use the Sample Some option.

It is also possible to set a distribution's sampling frequency to generate a new sample value at each Markov cycle/stage, during first-order trials and/or cohort/EV calculations. The DistForce( ) syntax can be used to resample more frequently.

Sampling during non-simulation, EV calculations

During non-simulation analyses, such as roll back and sensitivity analysis, a reference to a distribution normally returns the *mean* value every time it is referenced. The Dist() function can, however, override this behavior and return a randomly sampled value from the referenced distribution during any EV calculation.

To cause a random distribution sample to be returned by the Dist() function during expected value (EV) calculations, simply add a second parameter to the function with a value of 1. The formula Dist(*n*;1) will sample a new value from distribution number *n* each time the distribution is referenced in a tree calculation.

Or, use the DistForce(*n*) function instead, to do the same thing.

This also means that at each place in the tree where the distribution is referenced, a different sample value will be returned.

Correlating distributions

As described in the next section, distribution parameters can be defined using variables or formulas, instead of fixed numeric values. This can be used to some extent to correlate related distributions, but true correlation is a more involved process of determining covariances, etc.

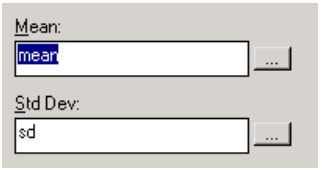
A more robust approach to correlation is possible, based on TreeAge Pro's support of user-defined Python functions (see Chapter 21). Python functions can use the multivariate normal and other distribution correlation tools found in the "RandomArray" and other Numerical Python extensions. Numerical Python requires a full installation of the open source Python software.

See the Numerical Python website:

http://numpy.sourceforge.net/numdoc/HTML/numdoc.htm

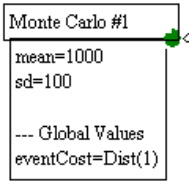
Defining distribution parameters non-numerically

Distribution parameters can be defined using variables or formulas, instead of fixed numeric values. This makes it easier to modify a distribution's parameters, and it also may help someone viewing the model understand the significance of a particular distribution.



Clicking on the ellipsis button to the right of a distribution parameter's text entry box in the Distribution Picker dialog will open an expression editor dialog. This dialog, like a variable definition window, makes it easier to set up a complex expression, including existing variables, functions, and even other distributions, to represent a distribution parameter.

Variables that are referenced in the parameter of a distribution should be defined default for the tree, at the root node. An error will be reported if TreeAge Pro cannot find a definition at the root node for the referenced variable.



It is possible to use the output of one distribution as a parameter in a second distribution, but TreeAge Pro requires that the dependent distribution have a higher numeric index than the input distribution. A distribution cannot reference itself as a parameter.

Approximating distribution parameters from statistical information

Some of the built-in distribution types supported in TreeAge Pro use parameters that may be difficult to find. For example, the Log-normal distribution is parameterized using the "mean of the logs" and the "standard deviation or error of the logs" – ***not*** to be confused with the log of the mean or standard deviation.

TreeAge can approximate some distributions' parameters from more commonly known statistical summary information. Distributions that support this approximation will make available a button labeled **Approximation**. Supported distributions include the real-number form of the Beta, the Log-normal, and the Gamma distribution

In the next section of this chapter, some of the details of these approximations are provided.

"Standard deviation" versus "standard error"

Some of the built-in distribution types supported in TreeAge Pro use parameters labeled "standard error". For example, the Log-normal distribution is parameterized using the "mean of the logs" and the "standard deviation or error of the logs" – ***not*** to be confused with the log of the mean or standard deviation.

Commonly-used distributions

Normal distribution

Formula: $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$

Domain: $-\infty < x < \infty$

Parameters: Mean $-\infty < \mu < \infty$, standard error/deviation $\sigma > 0$

Uniform distribution (real number form)

Formula:

$$f(x) = \begin{cases} 0 & x < LowX \text{ or } x > HighX \\ \frac{1}{HighX - LowX} & LowX \le x \le HighX \end{cases}$$

Domain: $LowX < x < HighX$

Parameters: Low X, High X

Uniform distribution (integer form)

Formula:

$$f(x) = \begin{cases} 0 & x < LowX \text{ or } x > HighX \\ & \text{or } x \text{ not an integer} \\ \frac{1}{HighX\text{-}LowX+1} & LowX \le x \le HighX \end{cases}$$

Domain: $LowX \le x \le HighX$

Parameters: Low X = real integer
High X = real integer > Low X

Triangular distribution

Formula:

$$f(x) = \begin{cases} 0 & x < LowX \text{ or } x > HighX \\ 2*(HighX - LowX)*\frac{x - LowX}{Mode - LowX} & LowX \le x \le Mode \\ 2*(HighX - LowX)*\left(1 - \frac{x - Mode}{HighX - Mode}\right) & Mode < x \le HighX \end{cases}$$

Domain: $LowX \le x \le HighX$

Parameters: Low X, Mode (likeliest), High X

Fractile distributions (10/50/90 et al)

Formula: Equal probability for Low X and High X (e.g., .30 or .25), remainder for Median, 0 for other values.

Domain: Low X, Median X, High X

Parameters: Low X, Median X, High X

More continuous distributions

Beta distribution (integer form)

Formula: $f(x) = \frac{(n-1)!}{(r-1)!(n-r-1)!} x^{r-1}(1-x)^{n-r-1}$

Domain: $0 < x < 1$

Parameters: $r > 0,\ n > r$

Details: Mean = r/n; r = occurences; n = population size

Beta distribution (real number form)

Formula: $f(x) = x^{(a-1)}(1-x)^{(b-1)} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}$

Domain: $0 < x < 1$

Parameters: $a > 0,\ b > 0$

Details: Mean = $a/(a+b)$

The parameters a and b can be approximated in TreeAge Pro from mean and standard deviation:

a = mean^2*(1-mean)/(se^2)

b = mean*(1-mean)/(se^2)-a

Dirichlet distribution (multivariate, normalized Beta)

Formula: $p_j = \frac{x_j}{\sum_{i=1}^{k} x_i}$ *where* $f(x_i) = \frac{x_i^{\alpha_i - 1} e^{-x_i}}{\Gamma(\alpha_i)}$

Domain: $0 \le p_j \le 1$ *where* $\sum_{i=1}^{k} p_i = 1.0$

Parameters: $\alpha_1, \alpha_2, \ldots \alpha_k$, $\beta = 1$

Details: A list of α_k specifies a ***k***-dimensional Dirichlet distribution. Each of ***k*** independent gamma variates is normalized by their sum. See Chapter 22 for usage details.

Chi distribution

Formula: $f(x) = \dfrac{2\left(\frac{n}{2}\right)^{\frac{n}{2}}}{\theta\,\Gamma\left(\frac{n}{2}\right)} x^{n-1} e^{\left(\frac{-nx^2}{2\theta}\right)}$

Domain: $x > 0$

Parameters: $n > 0,\ \theta > 0$

Chi-Squared distribution

Formula: $f(x) = \dfrac{x^{\left(\frac{n}{2}-1\right)} e^{-\frac{x}{2\theta}}}{(2\theta)^{\frac{n}{2}}\,\Gamma\left(\frac{n}{2}\right)}$

Domain: $x > 0$

Parameters: $n > 0,\ \theta > 0$

Erlang distribution

Formula: $f(x) = \dfrac{(k\lambda)^k x^{k-1}}{(k-1)!} e^{-k\lambda x}$

Domain: $x > 0$

Parameters: $k > 0,\ \lambda > 0$

Exponential distribution

Formula: $f(x) = \lambda e^{-\lambda x}$

Domain: $x > 0$

Parameters: $\lambda > 0$

Gamma distribution

Formula: $f(x) = \frac{\lambda^{\alpha} x^{(\alpha-1)}}{\Gamma(\alpha)} e^{-\beta x}$

Domain: $x > 0$

Parameters: $\alpha > 0,\ \lambda > 0$

Details: Mean = a / λ

The parameters a and λ can be approximated from a mean and standard deviation:

α = (mean^2)/(se^2)

λ = mean/(se^2)

Hyper-exponential distribution

Formula: $f(x) = 2\lambda p^2 e^{-2\lambda p x} + 2\lambda(1-p)^2 e^{-2\lambda(1-p)x}$

Domain: $x > 0$

Parameters: $\lambda > 0;\ 0 < p < 1$

Laplace distribution

Formula: $f(x) = \frac{1}{2b} e^{-\frac{|x-a|}{b}}$

Domain: $-\infty < x < \infty$

Parameters: $b > 0,\ -\infty < a < \infty$

Logistic distribution

Formula: $f(x) = \frac{ae^{-(ax+b)}}{\left(1+e^{-(ax+b)}\right)^2}$

Domain: $-\infty < x < \infty$

Parameters: $a > 0,\ -\infty < b < \infty$

Lognormal distribution

Formula: $f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{\left(\frac{-(\ln(x)-\mu)^2}{2\sigma^2}\right)}$

Domain: $x > 0$

Parameters: $\sigma > 0, \quad -\infty < \mu < \infty$

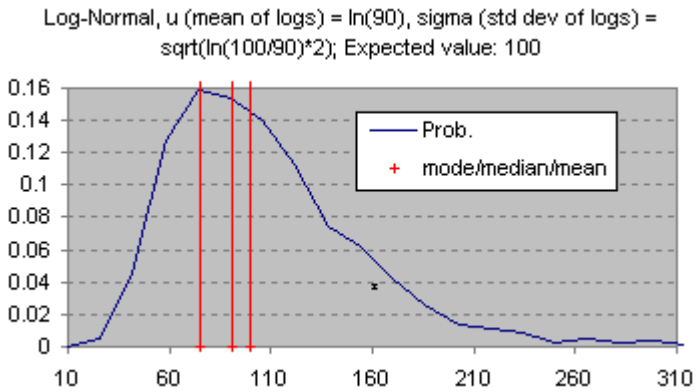
Details: The parameters μ and σ are the mean and standard deviation, respectively, from the normally distributed *natural logs* of x, and can be calculated in a variety of ways (if μ and σ are missing but the data are truly lognormally distributed):

$$\mu = ln(median)$$
$$\sigma = \sqrt{2 * ln(mean/median)}$$

In addition to these conversions, available in TreeAge Pro, also the following relationships::

$$\sigma = \sqrt{2 * ln(mean/mode)/3}$$
$$\sigma = \frac{ln(x_1/x_2)}{z_1 - z_2}$$
$$(mean = \bar{x}) > (median = \tilde{x}) > (mode = \hat{x})$$

The relationship of $\bar{x} > \tilde{x} > \hat{x}$ is illustrated in the distribution graph below (exported as an Excel chart from a TreeAge graph):



A well-documented, freeware Windows program supported by the Pacific Northwest National Laboratory, called LOGNORM4, can be used to calculate Lognormal distribution parameters from a variety of other statistics, including percentiles. Download and information at Strom DJ. 1994. LOGNORM4: Determining Parameters of Lognormal Distributions from Minimal Information. PNL-SA-22137. http://qecc.pnl.gov/LOGNORM4.htm.

Maxwell distribution

Formula: $f(x) = \frac{4x^2}{\alpha^3\sqrt{\pi}} e^{-\frac{x^2}{\alpha^2}}$

Domain: $-\infty < x < \infty$

Parameters: $\alpha > 0$

Rayleigh distribution

Formula: $f(x) = \frac{x}{a^2} e^{\left(\frac{-x^2}{2\alpha^2}\right)}$

Domain: $x > 0$

Parameters: $\alpha > 0$

Weibull distribution

Formula: $f(x) = \lambda k x^{k-1} e^{-\lambda x^k}$

Domain: $x > 0$

Parameters: $\lambda > 0, k > 0$

More discrete distributions

Binomial distribution

Formula: $P(x) = \binom{n}{x} p^x (1-p)^{n-x}$

Domain: $x = 0, 1, 2, \ldots, n$

Parameters: $0 < p < 1,\ n = 1, 2, 3, \ldots$

Poisson distribution

Formula: $P(x=i) = e^{-\lambda} \frac{\lambda^i}{i!}$

Domain: $i = 0, 1, 2, \ldots$

Parameters: $\lambda > 0$

Sampling from tables during Monte Carlo simulation

If it is not feasible or desirable to use one of TreeAge Pro's built-in distributions to represent the particular probability distribution you need, there are at least two ways to create custom sampling distributions. Both methods use a built-in distribution and a table.

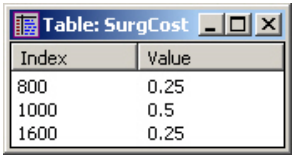
Creating a Table-type distribution

One way to sample values from a custom distribution is to create a new table in TreeAge Pro describing the distribution's discrete probability function (not the data set). This table can then be assigned to a Table distribution.

Each row of the table defines a distribution value (entered in the *index* column) and its probability (entered in the *value* of a table entry). The probabilities in the value column must total 1.0.

► **To create a Table distribution from an existing empirical data set:**

▷ Use the Variables and Tables dialog to create and a table and populate it with data that represents the custom distribution function.



For instance, you might create the simple table shown above for use as a distribution representing the cost of surgery. As you can see, the surgery is most likely to cost $1000, with approximate 10th and 90th percentile values of $800 and $1600, respectively.

▷ Next, choose **Values >** Distributions… to open the Distributions dialog, and click **New…** to add a distribution. In the Distribution Picker dialog, click on the **Table** distribution button.



▷ From the list of tables, select the name of the probability distribution table you created. Click OK to close the Distribution Picker dialog.

▷ In the Distribution Properties dialog, you can give the distribution a name and/or descriptive comment. Click OK to save the distribution.

To randomly sample from the Table distribution, simply reference the distribution in the standard way, as described in Chapter 16: use either the distribution's name or the Dist(*n*) function using the distribution's index.

Notes:

- Sample values will only be drawn from exact table entry indexes, regardless of which lookup method you specify. TreeAge will not interpolate in a Table distribution.
- The mean value of the Table distribution will be used as the distribution's expected value in non-Monte Carlo calculations.
- Chapter 16 describes a method for creating a Table distribution from one of the built-in, continuous or discrete distributions, using a custom table builder.

Using a distribution to lookup values in a table

There are situations where distributions cannot be easily represented using a standard distribution type, or a regular Table distribution as described above.

For instance, perhaps you have a table of age-dependent probabilities or costs that you want do probabilistic sensitivity analysis on. Or, you might have a set of observed parameter values that you want to bootstrap from, with each row's value given an equal probability (or even selected in order).

In these cases, you could populate the table and then use a separate distribution to sample row indexes and/or columns indexes.

For example, you could fill a table with observed data, numbered from 1 to N, and then pick values randomly from the table using a uniform distribution with a range equal to the range of table indexes. Use the integer form of the Uniform distribution, in this case (to return only integers in the index range).

► **To sample from a table using a Uniform distribution:**

▷ Paste your data set into a TreeAge Pro table, using consecutive integer indexes in the index column and the data set's values in the value column.

▷ In the tree, choose **Values >** Distributions… to open the Distributions dialog, and add a new distribution. In the Distribution Picker dialog, click on the Uniform button.

▷ For the low value, enter the lowest integer index from your table (i.e., 0 or 1). For the high value, enter the highest integer index from your table. Click OK to close the Distribution Picker dialog.

▷ In the Distribution Properties dialog, give the distribution a name and descriptive comment. Click OK to save the distribution, and then close the Distributions dialog.

The actual reference in a tree formula should look something like the following:

```
TableX[Dist(1)]
```

where "TableX" is the name of the custom distribution table, and inside the square brackets is the reference to the Uniform distribution. During a second-order simulation, the Uniform distribution will be resample within its range, causing different rows from TableX to be drawn randomly, with essentially equal likelihood if done correctly.

It may also be possible to pick rows from the table *in order* during a simulation, by using the automatic _sample (or _trial) counter in place of the Uniform distribution. The _sample counter corresponds to the current iteration of the simulation, incrementing by one at each resampling iteration.

To pick from a particular column in a multi-column table, simply add the appropriate column parameter to the table reference, such as:

```
TableX[Dist(1); 2]
```

To convert a time-dependent table of values for sampling, you might add additional columns that represent percentiles or bounds of each row. Then, you would use a distribution to sample a column index (i.e., between 2 and 3):

```
TableX[_stage; 2+Dist(1)]
```

Converting a distribution to a discrete table

In addition to sampling from parameterized distributions (e.g., Normal, Beta, etc.), TreeAge Pro can sample from discrete, custom distributions represented by rows of values and probabilities stored in a TreeAge Pro table.

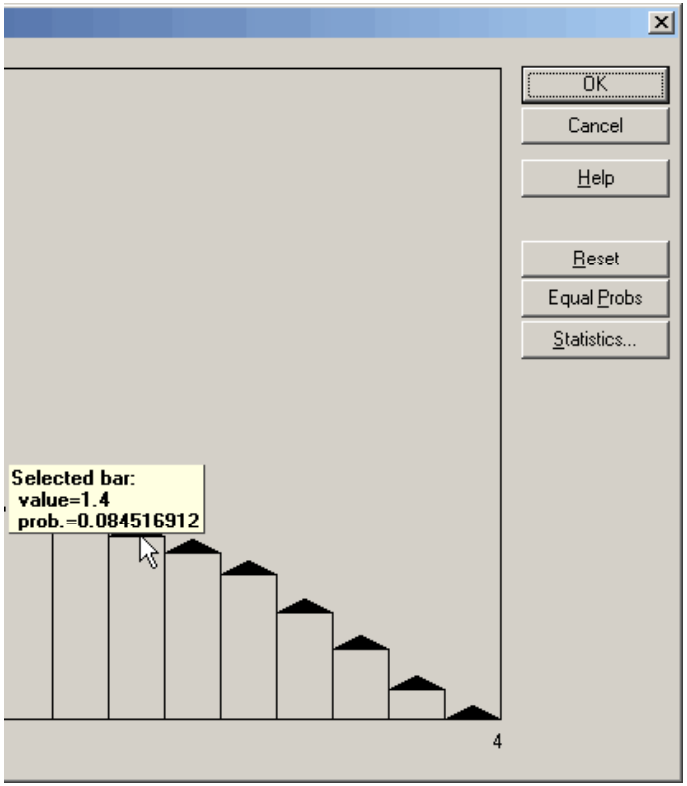
Release 1 of TreeAge Pro 2005 added a new, visual tool for constructing a custom distribution table, by dragging individual bars up or down in a distribution graph/histogram.

When selecting a distribution form/shape in the Distribution Picker dialog, the **Custom Table...** button provides access to the Custom Table Distribution builder. Based on the selected distribution shape and parameters, this tool can be used to customize the distribution, save the distribution in a TreeAge table, and assign the table to a distribution.

► **Creating the custom distribution table:**

- ▷ First, pick a distribution type in the Distribution Picker and assign its parameters.
- ▷ Click the Custom Table... button and in the Custom Table Distribution options dialog, specify a number of bars to divide the distribution into.
- ▷ In the Custom Table Distribution dialog, each bar corresponds to a row in a table. Drag a bar up or down to increase or decrease its relative probability.
- ▷ After you click OK, the table will be created.

The Distributions Picker will change the distribution to a Table distribution, pointing to the new table (or the updated table if you are editing an existing Table distribution).



A Table window will be open in the tree window, showing you the current entries in the underlying table.

Table: tCostDistrib

Index	Value
-3.8	0.000083198
-3.4	0.0064204
-3	0.009333746
-2.6	0.013758412
-2.2	0.02048918
-1.8	0.033303228
-1.4	0.051710421
-1	0.06330227
-0.6	0.086814589
-0.2	0.100127625
0.2	0.101666604
0.6	0.096448995
1	0.091166413
1.4	0.084516912

See Chapter 18 for further information on working with tables, table files, and packages.

Chapter 18

Creating and Using Tables

This chapter provides instructions on creating tables of numeric values for use in custom sampling distributions and other tree calculations.

In this chapter:

- Creating and editing tables 234
- Linking a table to an ODBC data source 238
- Referencing tables in formulas 241
- Packaged trees versus table files 242
- Managing TreeAge Pro's Tables directory 243
- The Tables Used report 244
- Managing tables using the Excel/COM module 244

Creating and editing tables

TreeAge Pro can store indexed tables of numeric values that represent parameters in your model. Here are some basic facts about tables:

- Tables are created initially as separate files available to all trees, stored in the TreeAge Pro\Tables subdirectory. However, a tree can be saved as a package file which will contain copies of referenced tables.
- Every table has an index column and a value column. A table can have additional value columns (up to 512) and from one row of values up to tens of thousands of rows.
- Each row's index value must be unique, but indexes are *not* required to be consecutive integers. TreeAge automatically numbers the value columns (starts at 1).
- Tables can linearly interpolate values for missing indexes/rows, as well as columns.
- Only numeric values (no variable names or formulas) can be entered in a table.
- In Markov models, tables are often used to represent probabilities that vary over time (or other dimensions). See Chapter 34.
- A table can represent a parameter's empirical probability distribution, to sample from during Monte Carlo simulation. (See Chapter 16 for details on sampling from tables and other distribution types.)
- Tables created in TreeAge are ***not*** text files. Values can be loaded into a table in a variety of ways, including: paste tab-delimited text; edit via the Excel add-in; query an ODBC database; manually enter index-value pairs; convert a graph.

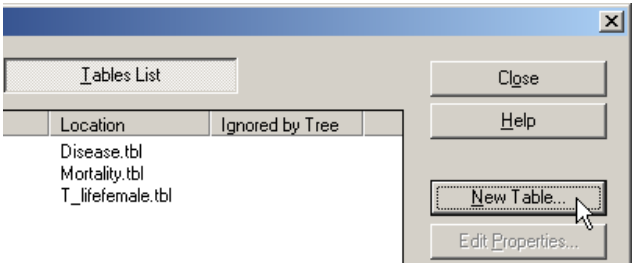
Creating an empty table file

In order to enter values in a table, you must first create an empty table file, giving it a variable-type name and a file name.

► **To create a new table:**

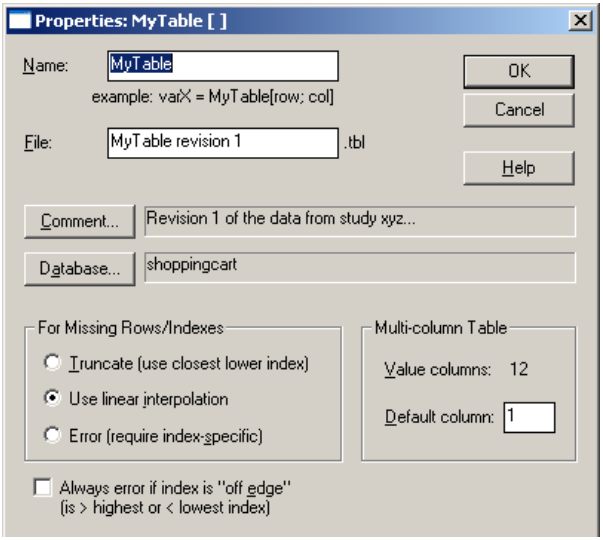
▷ In a tree window, choose **Values >** Variables and Tables…. In the Variables and Tables dialog, click on the Tables List button.

▷ Click on the **New Table…** button.



The Table Properties window, shown below, will appear, prompting for basic table properties.

▷ In the Name box, enter the name you want to use in formulas referencing values in the table. Follow variable name rules: up to 32 characters; no spaces or punctuation other than the underscore ("_").



- ▷ You can leave the Filename box empty unless you want to save the file with a different filename than the table name.
- ▷ If your table skips some row indexes (or columns) that formulas may reference (e.g., you want to interpolate between ages 65 and 75), choose an appropriate lookup method to handle missing rows or columns. Lookup method options are explained later in this chapter.
- ▷ Add a descriptive comment, if desired, by clicking the **Comment…** button.
- ▷ Click OK to create the empty table file.

The empty table will be added to the tables list. (Also, if you look in TreeAge Pro's Tables subdirectory, you will find a new *.tbl file with the filename specified.)

► **To change the properties of a table:**

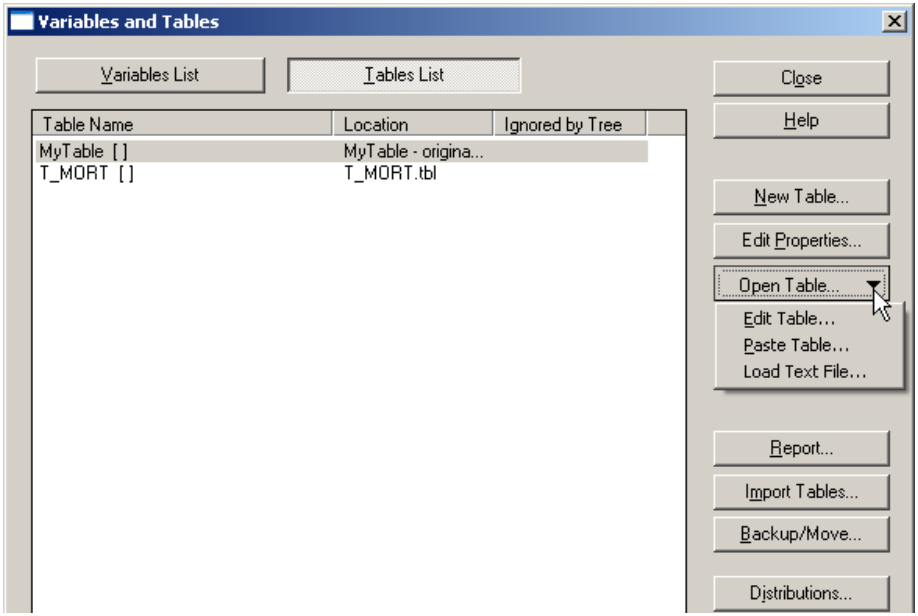
- ▷ In the Variables and Tables dialog, select the Tables List.
- ▷ Double-click on the name of the table in the list, or click the **Edit Properties** button.
- ▷ Make changes to the table properties and then click OK to save changes.

Entering values in a table

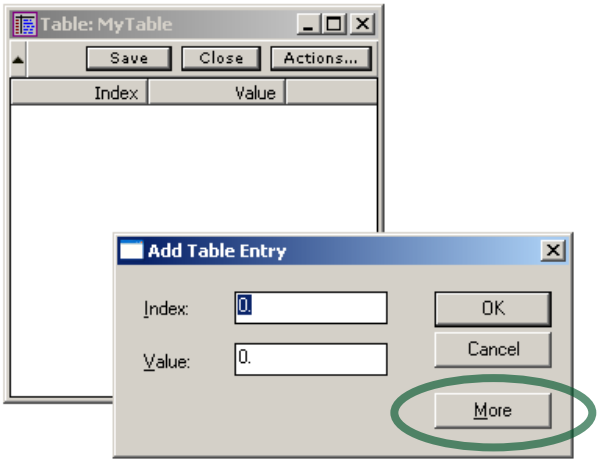
Tables created in TreeAge Pro are not text files, and can only be edited by TreeAge Pro. To populate a simple table, a Table window can be opened allowing you to add, edit, or delete one index-value pair at a time. The window also has options for copying/loading tab-delimited data from a spreadsheet, text file, or other source.

► **To open a Table editing window:**

- ▷ In the Variables and Tables dialog's Tables List, select the name of a table in the list. Then click the **Open Table…** button and choose Edit Table….



A Table window will open showing the contents of the table. If the table is empty, an Add Table Entry dialog will automatically be opened. If your table is small, requiring only a few index-value pairs, use this dialog to quickly populate the table.



► **To enter single index-value table pairs:**

▷ In the Table window, click **Actions…** and choose Add Entry… (if the entry dialog is not already open).

▷ To enter multiple index-value pairs, click the **More** button. Click OK to update the Table window.

The Table window's **Actions…** menu also includes Delete and Edit commands for selected rows/pairs.

Limitations of the Table window

The Add and Edit Table Entry dialogs will not work with tables having more than one column of values. If a table has multiple value columns, many rows, or must be updated periodically, either:

- Use a spreadsheet or other text editor to set up the table, and then copy it as tabbed text into the Table window; or
- Set the table's ODBC database properties, to load values from a database query.

These options are described in the next sections.

Loading/copying tab-delimited table data

To modify the values in an existing TreeAge Pro table, you can first copy it into a spreadsheet or text file for editing.

► **To copy data from an existing table:**

▷ Open a Table window in TreeAge Pro, as described above. Then choose the Copy Table command from either the Actions… or Table menu. The table values will be copied onto the Windows clipboard in tab-delimited format.

▷ Switch to your target application, such as Excel, paste the table values, and make required changes.

Once you are ready to copy a range of spreadsheet cell values back into the Table window, take note of

	A	B	C	D	E
1	Index	baseline (c	low (c2)	high (c3)	
2	55	0.5	0.3	0.7	
3	65	0.55	0.3	0.8	
4	75	0.65	0.35	0.85	
5					

the following requirements:

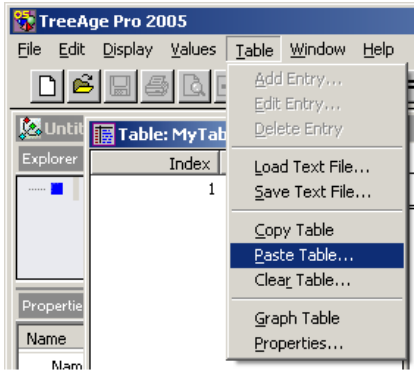
- A column title row is *not* required. If one is included, however, the title of the index column *must* start with the word "Index". Value column titles have no restrictions.
- The selected range of cells should use either General, Text, or Number formatting (avoid accounting formats using parentheses for negatives numbers).
- The copied range should not include empty columns or rows.
- Instead of copying very large tables, it may be preferable to save the source document as a tab-delimited text file and load it into a Table window using the **Table >** Load Text File… command.

Note that the Paste Table and Load Text File commands first remove existing rows from the table. (In other words, they clear the table using the **Actions** > Clear Table command.)

► **To paste tab-delimited data into a table:**

▷ Select and copy the table in the source spreadsheet or text document.

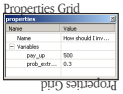
▷ In TreeAge Pro, select the table in Tables list, click **Open Table…** and choose Paste Table. (Or, if the Table window is open, choose Paste Table from either the Actions or Table menu.)



If the Paste command is not available, or does not work, see the bulleted list of suggestions above on proper cell formatting in the source document. If all else fails, try first pasting the spreadsheet data into a plain text file, to remove formatting, and then copy the text file data into TreeAge.

Table window/editing tips:

- To open a Table editing window from the Variables and Tables dialog, press ALT+ENTER with a table selected, or ALT+double-click on the table in the list.
- The menu commands in the Actions… toolbar button are also found under the **Table** menu in the TreeAge menu bar.
- The tree's properties grid can also be used to edit existing index-value pairs in a table file. Simply click on the root node of a tree, expand the list of available table files, and select the desired table.



- Excel module users: With the optional **Excel module**, the Tables List (as well as the Variables List and Distributions List) includes an **Edit in Excel…** command.



  This allows tables to be transferred directly to Excel (instead of copying/pasting). The Excel module's TreeAge add-in menu includes a related Add or Update Table… command. See Chapter 28 for details.

- Saving changes: When you make changes to the values using the Table window, TreeAge keeps a backup of the original table contents. When you close the Table window (e.g., click the OK button), TreeAge will ask you to confirm you want to save the changes.
- Read the section on Packages later in this chapter for information on the differences between editing global tables (i.e., table files) and editing tables stored with a packaged tree.

Linking a table to an ODBC data source

TreeAge Pro 2007 adds the option to link a table to an existing database or other data source that has an ODBC driver. Linking a table to a database is an easy way to make table updates automatic.

> ***Note:*** This section covers the basic elements of connecting a TreeAge table to a data source. For more details on this topic, press the Help button in the database properties dialog.

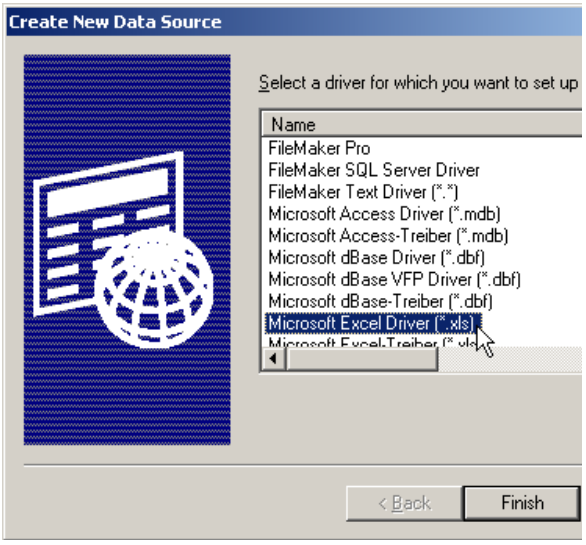
TreeAge Pro tables can use ODBC to access most commercial and open-source database formats, including MySQL, SQL Server, Access, Firebird, as well as Excel spreadsheets, text-format files, and XML files. A standard SQL SELECT query is used to select columns and rows from a table in the data source, which then replace the contents of the target TreeAge Pro table.

Using ODBC data sources in Windows

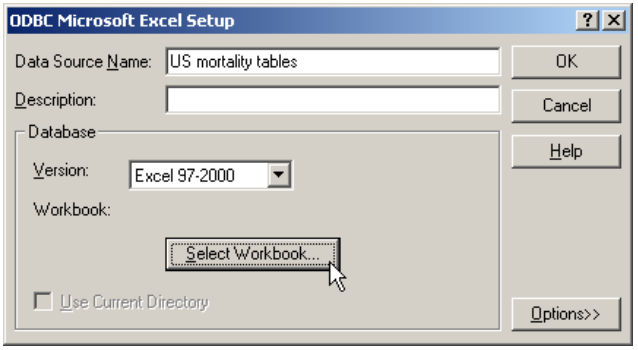
Once you have a data source, you can make it accessible in Windows via ODBC:

► **To create a DSN (Data Source Name):**

▷ To open the DSN list, go to **START > Run…** and open the program `odbcad32`. Or, open an Explorer window, go to Control Panels > Administrative Tools > Data Sources.



▷ Add a DSN for your data source.



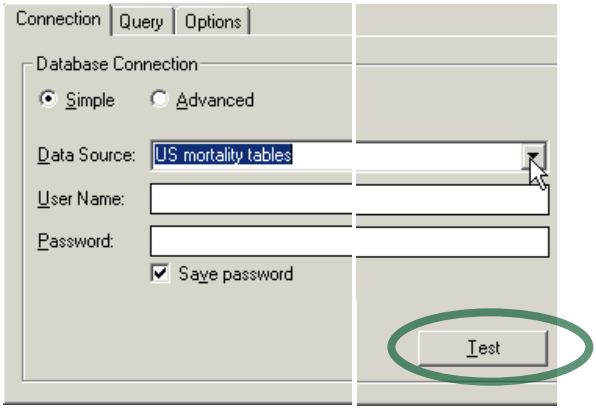
Setting a table's data source properties

Your TreeAge table can be linked to a data source via the ODBC DSN you define.

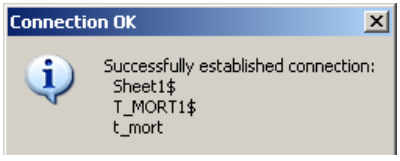
► **To set the data source connection properties for a table:**

▷ In a TreeAge Pro tree window, choose **Values >** Variables and Tables…, and click on the **Tables List** button. Select a table, click on the **Edit Properties…** button, and then click on the **Database…** button.

▷ On the Connection tab, select the DSN from the list of available sources. Specify a user name and password, if required.



▷ Click the **Test** button to verify that TreeAge can connect to the database, in which case the list of available tables in the data source is shown.

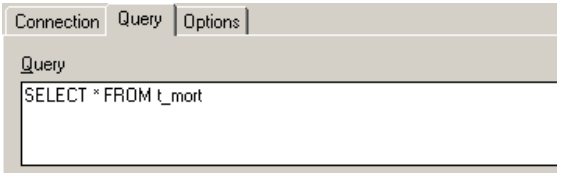


In a spreadsheet, worksheets and named ranges will be listed. A regular database, on the other hand, will provide the list of named tables. The query you create in the next step will refer to one of these tables.

A simple query like `SELECT * FROM t_mort` can be used to retrieve all rows and columns from the specified table, for example a range of spreadsheet cells named t_mort.

► **To query a table in the data source:**

▷ Switch to the Query tab and type an SQL SELECT query to send to the data source.



The SELECT query syntax will be very similar for all data sources. For example, the * character in an SQL query means "retrieve all columns from the table". Another standard in the SQL syntax is the use of a WHERE clause to pick a specific set of rows.

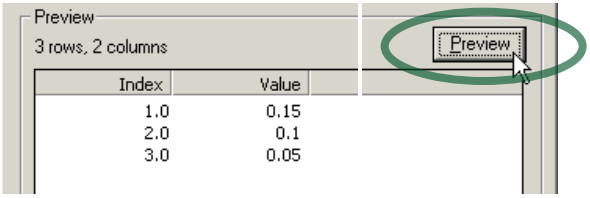
For example, you might specify:

```
… t_mort WHERE study=5
```

to retrieve only a matching subset of records/rows.

Refer to a general guide to writing SQL queries for detailed help (for example, the JetSQL guide at the http://www.devguru.com/ web site).

▷ Click **Preview** to confirm the query syntax is valid, and to see a sample ***recordset*** (i.e., the results of the query). .



TreeAge will use the recordset as follows:

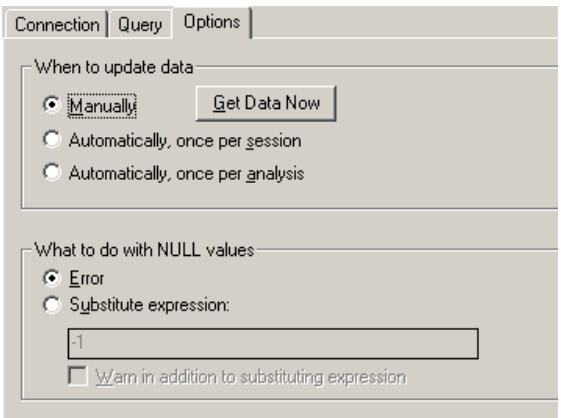
- Each row in the returned recordset will become a row in the TreeAge Pro table.
- The first column in the recordset can be used as the table's index column, in which case this column should contain unique numeric values identifying each row.
- Alternatively, you can select the option labeled Generate index column to have TreeAge Pro create its own, uniformly-increasing index values:



Once a valid data source query has been created, you should specify when to run the query and update the TreeAge table:

► **To specify the table update frequency:**

▷ Switch to the Options tab and specify when the table should be requeried. The options are covered in the next section.



The What to do with NULL values options determine how TreeAge handles empty cells in numeric columns returned in the recordset.

► **To avoid errors due to NULL values:**

▷ Select the option Substitute expression and enter a default value to replace NULLs.

When to requery data sources

The table's data source options include three settings for when to requery and update the table:

- manually update the table using the **Get Data Now** button here, or in the Databases dialog described below;
- automatically update the table once per TreeAge Pro session, the first time the table contents are needed; or
- automatically update the table prior to each analysis.

If your table's data source does not change frequently, one of the first two settings will probably be appropriate.

If you are retrieving a large recordset or running complex queries, using the third option, updating prior to each analysis, may cause unnecessary delay. This third option will probably only be necessary in special circumstances.

An additional option for requerying a table's data source is to is to force an update at any time by evaluating the special function:

```
Command("tables";"*";"requery")
```

See Functions, Appendix E, for more on the Command function.

In addition to manually requerying the table using the Get Data Now button in the table's data source Options tab, you can also do this via the Databases list.

► **To manually requery tables:**

▷ In the tree, choose **Values >** Databases…. Select the database tables to requery, and click **Get Data Now**.

Referencing tables in formulas

There are two basic ways in which tables are used in trees:

- If a table is used as an empirical sampling distribution, then you simply select its name from a list of tables when setting up the distribution, as described later in this chapter.
- In order to lookup a value in the table from a payoff, probability, or other calculation, however, you will need to use the proper syntax.

Table syntax

In a table composed of index-value pairs (i.e., one value column), a value can be retrieved with the following syntax:

```
TableName[ index ]
```

The table's variable-type name is followed immediately by square brackets containing an index used to pick a table row. The index expression can be a number, a variable, or even another table reference.

If a table has more than one value column, specify the number of the value column to pick from (after the row index argument):

```
TableName[ index; column # ]
```

Table lookup methods

During tree calculations, if a formula references a table using an index value that exactly matches a row index in the table, TreeAge simply returns the value from the appropriate column in that row. However, when a reference is made to a *non-existent* row/index, the table's missing row lookup method determines what is returned.

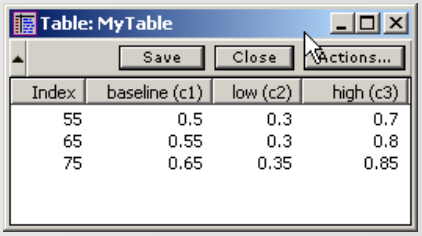
The missing row/column lookup methods are:

- **Truncation** – If rows with lower indexes exist, returns the value from the row with the highest index less than or equal to the requested index; otherwise, returns the value from the first row (i.e., the lowest index value).
- **Interpolation (default)** – Returns a value calculated by linear interpolation between existing indexes.
- **Index-Specific** – Report an error.

Each table also has an option "Index off edge is error." If this option is left ***unselected***, a table that uses truncation or interpolation will allow references to indexes above or below the table's actual range. In these cases, the table will return the value associated with the closest existing row or column. If this option is checked, on the other hand, an error will be reported.

Interpolating between columns

Starting with TreeAge Pro 2007, tables can simultaneously interpolate between columns as well as between rows. For example, in the table below, the reference MyTable[60; 3] will return 0.75, while the reference MyTable[60; 2.5].will return 0.525, or the average of 0.3 and 0.75.



Packaged tables versus table files

Each table that your create is stored initially as a separate file (with a ".tbl" extension) in TreeAge Pro's Tables subdirectory. Once it is created, a table file is then available to all TreeAge Pro trees. In many cases, this default system of storing tables has disadvantages, particularly if you:

- work on multiple projects each with its own tables
- work on projects at multiple computers
- share trees and tables with other users

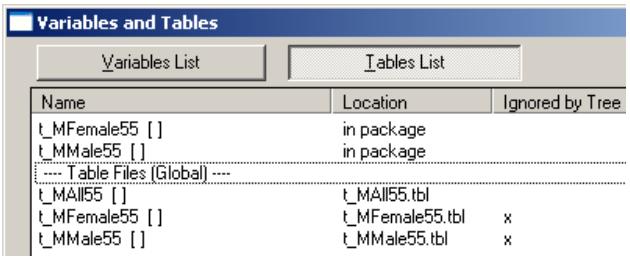
To deal with these situations, TreeAge Pro uses *packages*. A package file contains a tree and the set of tables used in the tree.

► **To save a tree as a package:**

▷ Choose **File >** Save As…, and select "Package Files (*.pkg)" from the "Save as type" list.

When a tree is saved as a package, TreeAge Pro determines which *table files* are referenced by the tree and packages them along with the tree. Only tables referenced by the tree and found in TreeAge Pro's Tables subdirectory will be packaged.

Assuming no table files are missing to begin with, the newly-created package will include all tables required by the tree. The package file can then be sent to another computer with TreeAge Pro and opened. Package files are treated just like tree files by version 2005 and later. They are simply opened, not imported.



Every time the **Save As** command is used to save a package, TreeAge Pro creates a clean list of required tables, removing from the package any tables that are no longer referenced. The regular **Save** command, on the other hand, only attempts to add required tables to the existing package; it does not remove unreferenced tables.

Once a package has been saved, the Variables and Tables dialog divides the tables lists into two parts: tables stored in the package; and original copies of table files found in the Tables subdirectory.

Notes on packaged tables:

- Tables stored in the package take precedence over global table files. If TreeAge Pro is able to locate a required table in the package, it will ignore a table *file* with the same name. If the package is missing a required table, TreeAge Pro will look for it in the list of global table files.
- In the Variables and Tables dialog, table files marked with an "x" in the Ignored by Tree column are ignored because a table of the same name is found in the package.
- Changes to package's tables are saved into the package only. (Changes are not saved to older copies in the global Tables files directory.)
- If you change the name of a table in a package, formulas in the tree that reference the table will be updated (as when a variable name is changed). Formulas are not updated if you change the name of a global table file.
- The Export command can also be used in TreeAge Pro to make a *protected package* if you want to password protect a tree.
- The Tables Used report (see later in this chapter) can be used to determine if any tables required by a tree are missing from the Tables directory, and so cannot be included in a package.

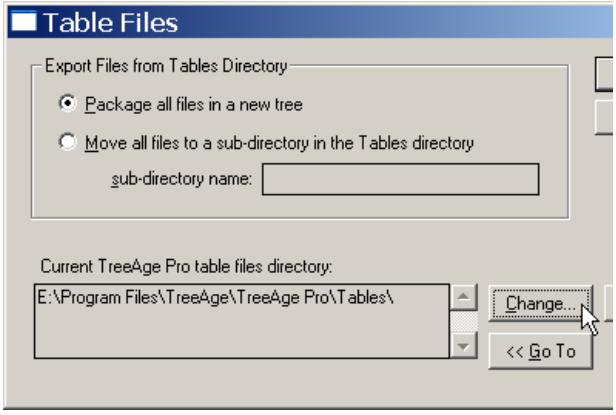
Managing TreeAge Pro's Tables directory

The default location of the global Tables directory is in the TreeAge Pro application directory. Using the **Backup/Move** button in the Variables and Tables dialog, it is possible to use a different directory.

► **To change the tables directory:**

▷ In the Variables and Tables dialog, click the **Backup/Move…** button to open the Table Files dialog.

▷ Click the **Change…** button.


Table Files
Export Files from Tables Directory
Package all files in a new tree
Move all files to a sub-directory in the Tables directory
sub-directory name:
Current TreeAge Pro table files directory:
E:\Program Files\TreeAge\TreeAge Pro\Tables\
Change...
<< Go To

It is a good idea to avoid deleting non-trivial tables without creating a backup. Instead, use the **Backup/Move** button to package all of your table files into a backup package.

► **To backup all table files in a new package:**

▷ In the Variables and Tables dialog, click the **Backup/Move…** button.

▷ Select the default option, Package all files in a new tree, and click OK. Select a location to save the new package.

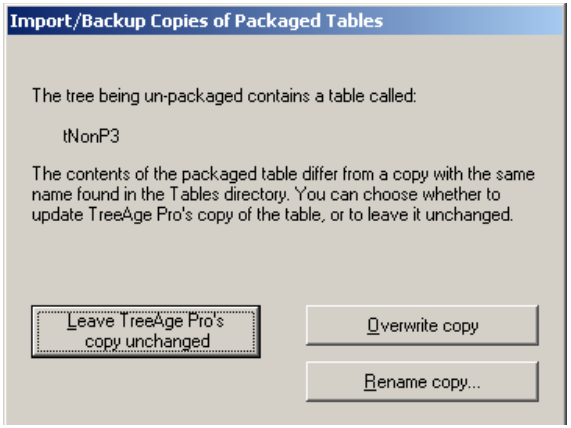
► **To move all table files into a new, backup subdirectory (in the Tables directory):**

▷ In the Table Files dialog, choose Move all files…. Enter the name of a new backup directory to create in TreeAge Pro's Tables subdirectory. Choosing a new directory will ensure that the move will occur successfully. TreeAge Pro will not allow existing table files in the specified directory to be overwritten.

Using the Save As command to save a package as a regular tree again **will not extract copies** of the tables to TreeAge Pro's Tables subdirectory. Use the **Import** button to unpack the tables from a package and store them as global table files.

► **To import a package's tables as table files:**

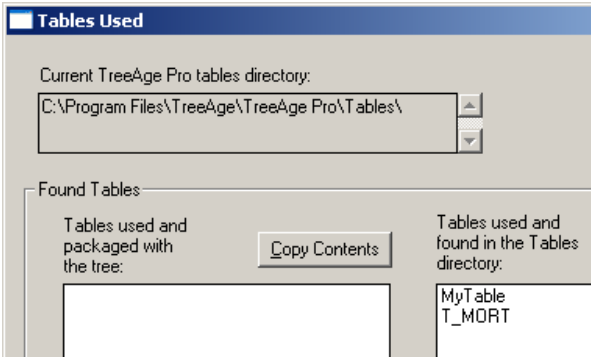
▷ In any open tree or package, choose **Values >** Variables and Tables…, click the **Import…** button, and select a package file from which to import tables.


Import/Backup Copies of Packaged Tables
The tree being un-packaged contains a table called:
tNonP3
The contents of the packaged table differ from a copy with the same name found in the Tables directory. You can choose whether to update TreeAge Pro's copy of the table, or to leave it unchanged.
Leave TreeAge Pro's copy unchanged
Overwrite copy
Rename copy...

▷ During the import process, if there is an existing table file that differs from a table in the package, TreeAge Pro will ask you to choose whether to update the disk copy or not.

The Tables Used report

For a tree to work correctly on a new computer, the required table (*.tbl) files must be copied along with the tree file or included in a package file. Before sending someone a packaged model, you can quickly create a text report listing the tables referenced by a tree, including their contents and properties, use the **Values >** Tables Used… report. This report can also be opened from the Variables and Tables dialog.



The internal (not file) names of the required tables for a particular tree are reported directly in the Tables Used dialog. The report also includes a second listing of the names of tables referenced in the tree, but missing from the Tables subdirectory (or package).

If you receive a model sent from another computer, but are unable to calculate it because of errors related to missing tables, the Tables Used report will allow you to generate a list of both found and missing tables.

Copying the information about found tables to the clipboard will include the values and lookup methods of these tables.

Managing tables using the Excel module

Users with the Excel/COM module can utilize the TreeAge add-in menu in Excel to update TreeAge with changes made to tables stored in a spreadsheet. See Chapter 28.

It is also possible to use the TreeAgeProLib to script changes to tables. See Chapter 30.

Tools and Options for Complex Models

The chapters in Part IV cover a variety of intermediate and advanced software features. These include such topics as creating complex cost formulas, linking to spreadsheets, using utility functions, performing Bayes' revision, and sharing your models using custom interfaces .

In this part:

- Chapter 19, "Storing Analyses and Analysis Sequences"
- Chapter 20, "Dynamic Linking to Excel"
- Chapter 21, "Large Trees: Using Clones, the Explorer View, and Links"
- Chapter 22, "Advanced Chance Node Techniques and Options"
- Chapter 23, "Bayes' Revision"
- Chapter 24, "Risk Preference and Utility Functions"

CHAPTER 19

USING STORED ANALYSES AND ANALYSIS SEQUENCES

All of TreeAge Pro's graphs can be saved and reopened for later viewing. Alternatively, the settings used to perform any analysis in TreeAge Pro can be stored. This chapter provides instructions on saving and using stored analyses. It also describes how to create simple, stored analysis sequences that, for example, run series of simulations or sensitivity analyses.

For more complicated tasks, such as batching lengthy analyses, automating export of analyses to spreadsheets, or retrieving analysis parameters from a database or spreadsheet, TreeAge Pro with the Excel/COM module can be scripted using, for example, Excel macros. See Chapter 30 for details.

In this chapter:

- Using stored analyses 248
- Sequencing stored analyses 251
- Stored analyses in a run-time custom interface 256

Using stored analyses

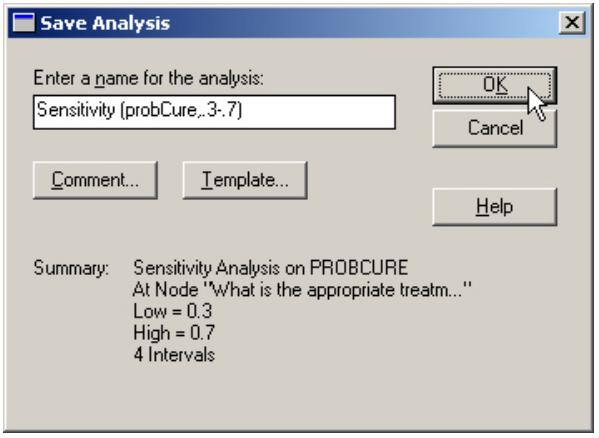
It is possible to save and reuse the parameters for most of the analyses available in the Analysis menu. This includes Monte Carlo simulation and all forms of sensitivity analysis. Analyses that cannot be stored include Graph Risk Preference Function, Show Optimal Path, Verify Probabilities, and Roll Back.

The first step is to run a regular analysis. (Note that if you are storing a Monte Carlo simulation for reuse, you do not have to let the simulation run to completion, just long enough to be able to calculate statistics on the partially complete simulation.)

► **To store the parameters of an analysis:**

▷ Open a tree and perform an analysis. After you have reviewed the results, close the output window, graph, or dialog, and switch back to the tree window.

▷ Choose **Analysis >** Storage > Save Last. You must choose this menu item before doing another analysis, as TreeAge Pro has temporary storage for only one set of analysis parameters.



▷ Enter a short, descriptive name for your analysis. This name is for your own reference (and also for other user's of your model). You may enter a longer description of the analysis by clicking the Comment… button.

▷ Press ENTER or click OK.

The instructions TreeAge Pro needs to perform the identical analysis are stored in the tree (the results of the analysis are not).

Running stored analyses

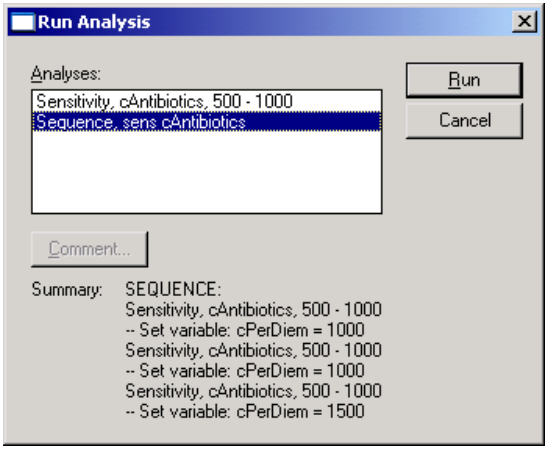
Once an analysis is stored, it is saved with the tree, and can be run as long as no significant structural changes are made where the analysis was originally run.

► **To run a previously stored analysis:**

▷ Open the tree in which you have stored an analysis.

▷ Choose **Analysis >** Storage > Run Old Analysis.

▷ Select the desired analysis from the list of stored analyses. TreeAge Pro displays a summary of the analysis. If you have entered a comment for the analysis, you may view it by pressing the Comment button.



▷ Click the **Run** button to start the analysis.

Many analyses that can be stored are dependent on the structure and variable definitions of your tree. For instance, the location (relative to the root node) of the node at which you originally performed a sensitivity analysis must be unchanged from when you stored the analysis, and all analyzed variable definitions must be present in the same locations. If TreeAge Pro is unable to reconcile the different structures, it will not run the analysis.

Editing, copying, and deleting stored analyses

The Maintain Analyses dialog can be used to change the name or comment of a stored analysis, delete or copy an analysis, assign a graph template, or create a sequence of stored analyses.

► **To edit the list of stored analyses (or sequences) in a tree:**

▷ Choose **Analysis >** Storage > Maintain Analyses to open the Maintain Analyses dialog.

► **To copy a stored analysis (or sequence) to another tree:**

▷ Choose **Analysis >** Storage > Maintain Analyses to open the Maintain Analyses dialog.

▷ From the list, select the analysis you wish to duplicate, and click the **Copy** button. Close the Maintain Analyses dialog.

▷ Switch to the second tree (which must be identical to the first tree in all respects material to the stored analysis; see Running stored analyses, above)

▷ Choose **Analysis >** Storage > Maintain Analyses in the second tree, and click the **Paste** button in the dialog.

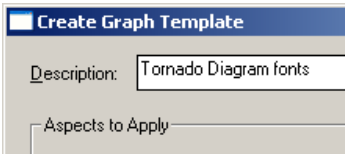
Using graph templates with stored analyses

As described in Chapter 6, a graph template is an abstract of the visual, settings used by a graph, including fonts, certain text labels, and numeric formatting. Normally, a graph template is applied manually to a graph afterit is created. It is also possible to assign a graph template to any stored analysis that automatically generates a graph.

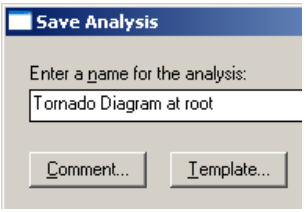
See Chapter 6 for general information on graph templates. This section assumes you are familiar with creating and using graph templates, and only describes the use of templates with stored analyses.

► **To store a graph template with an analysis:**

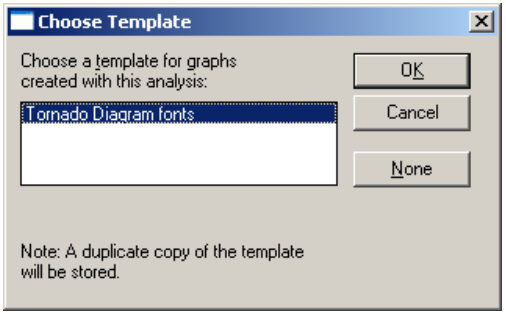
▷ Create a template from a graph window.



▷ Then, either in the initial Save Analysis dialog or, later, when editing the stored analysis from the Maintain Analyses dialog, click the **Template…** button.



▷ Select a graph template from the list, and click OK to attach the template to the stored analysis.



A separate copy of the graph template is included with the stored analysis. If you change or delete the original graph template, the copy in the stored analysis will not be affected.

To change the template stored with an analysis, reopen the Choose Template dialog from the Maintain Analysis dialog, and select another template or detach the current template by clicking the None button.

Sequencing stored analyses

In TreeAge Pro, sequences of stored analyses can be defined and stored, like a batched analysis, for future use. Analysis sequences can greatly simply tasks like batch processing a series of lengthy analyses, such as Monte Carlo simulations. Sequences can also be used if you frequently do manual kinds of sensitivity analyses, by repeating any of TreeAge Pro's built-in analyses over a range of variable values.

Creating an analysis sequence

The next four screenshots provide a simple illustration of how the sequencing feature can be used to automate running a series of separate Monte Carlo simulations, each using a different value of a variable.

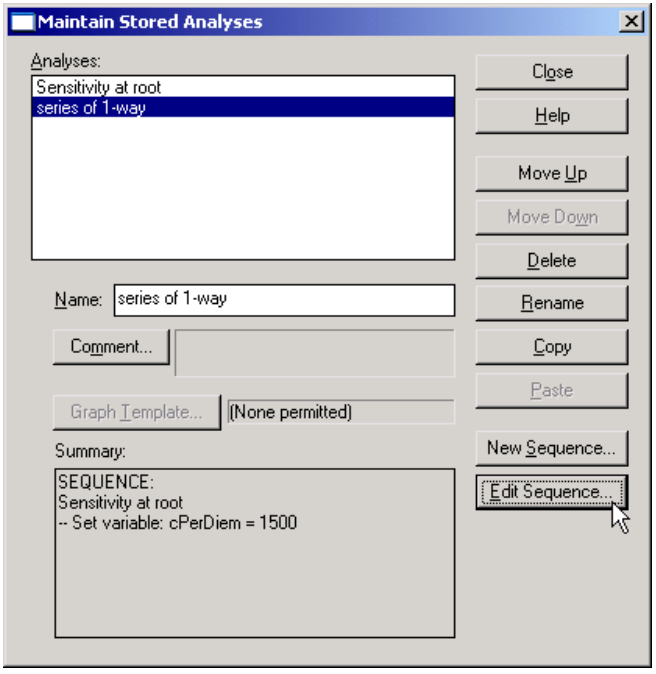
► **To create a sequence of stored analyses:**

▷ Run an analysis that can be stored, and then save it as a stored analysis in the tree, as described earlier in this chapter.

▷ After storing the analysis, choose **Analysis >** Storage > Maintain Analyses. In the Maintain Analyses dialog, click the button labeled **New Sequence…**.

▷ In the Sequence Stored Analyses dialog, enter a name for the new sequence.

The stored analyses listed in the Maintain Analyses dialog will also show up in the list on the left of the Sequence Stored Analyses dialog, shown on the following page.



Each step in a sequence is either: (1) a stored analysis; (2) a definition of a variable at the root node; or (3) a script command.

► **To add a stored analysis to the sequence:**

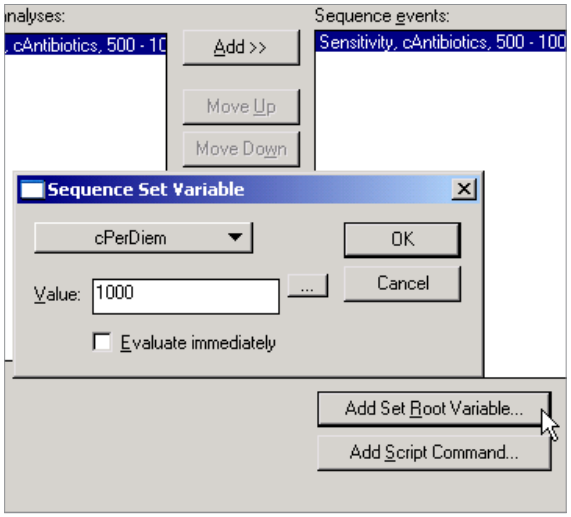
▷ Select the analysis from the list on the left and click the **Add >>** button to insert the analysis into the sequence list on the right.

Simply repeat the add operation to insert additional iterations of this, or another, stored analysis at the end of the sequence.

► **To change the value of a variable:**

▷ Select an analysis in the sequence, and click the **Add Set Root Variable…** button.

▷ In the dialog, select a variable from the list, and enter a value or expression with which to redefine the variable. Click OK to add the variable definition to the end of the sequence.



The definition is performed at the root node. Thus, if the variable was already defined at the root node, that definition will be overwritten by the definition specified during a sequence.

Variable definitions need not be numeric; they can reference functions and other variables defined at the root node. For more details, see the section on setting variable definitions.

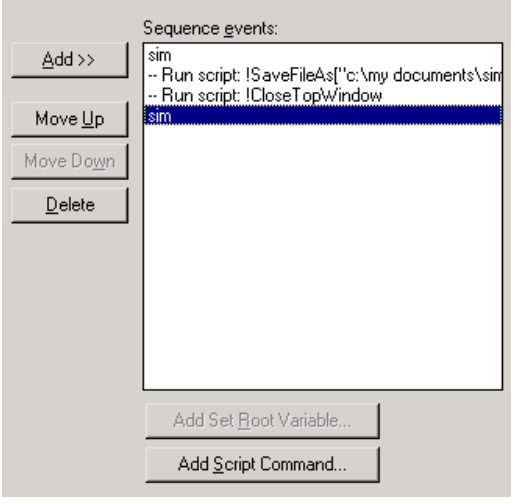
It is also possible to add script commands steps to the sequence. For example, new Monte Carlo output windows could be saved and closed before starting the next simulation in the sequence.

► **To run a script command in the sequence:**

▷ Select an analysis in the sequence, and click the **Add Script Command…** button.

To save and close a simulation window, two commands could be used, as in the example shown below:

!SaveFileAs["*path-and-filename*.mcs"]
!CloseTopWindow



See Appendix F for more information on using TreeAge Script commands.

► **To delete an analysis or definition from a sequence:**

▷ Select an analysis or definition from the list of sequence steps, and click **Delete**.

► **To reorder steps in the sequence:**

▷ Select a step list of sequence events, and click the **Move Up** or **Move Down** button.

Once you are certain you have a sequence of analyses and other steps, you can store the new sequence. (***Note:*** The sequence can be edited later, after it has been stored.)

► **To save the sequence in the tree:**

▷ Click OK to close the Sequence Stored Analyses dialog. The sequence will be stored in the tree, and its name added to the list of stored analyses.

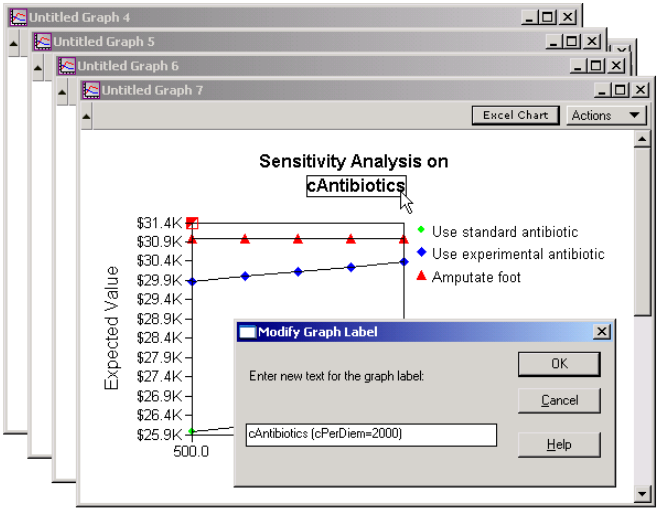
Once saved, the stored sequence is run the same way a regular stored analysis is run.

► **To run a stored analysis sequence:**

▷ Choose **Analysis >** Storage > Run Old Analysis.

▷ Select the sequence from the list of stored analyses, and click the Run button.

If a series of one-way sensitivity analyses are run, for example, you will see new graph windows appear as each analysis completes.



Notes:

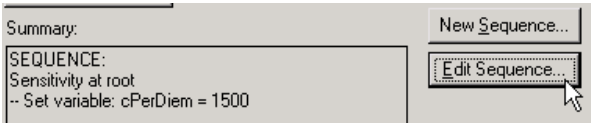
- Existing sequences cannot be used in a new sequence, and will not be included in the list of available analyses.
- It is possible to modify the underlying stored analyses that are included in a sequence, after creating the sequence. Delete the original stored analysis used in the sequence, and save a new stored analysis, using precisely the same name.

Editing stored sequence

Existing sequences can be modified, in order to add or remove steps, using the Add/Delete/Move buttons in the Sequence Stored Analysis dialog.

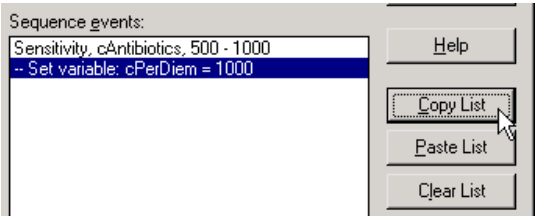
► **To edit a sequence in the Sequence dialog:**

▷ Select the sequence from the list of stored analyses, and click the **Edit Sequence…** button.



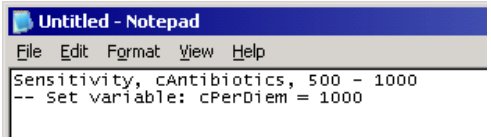
Sequences can also be modified by copying steps as text to a text editor. Lines can be added or removed from the sequence using the text editor, and then the updated sequence can be pasted back into the Sequence list, to add or replace current steps in the sequence.

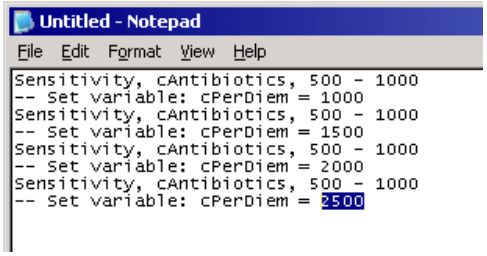
► **To edit a sequence in a text editor:**



▷ In the Sequence editor, click the Copy List button.

▷ Switch to the text editor program, paste the list, and make the required changes.

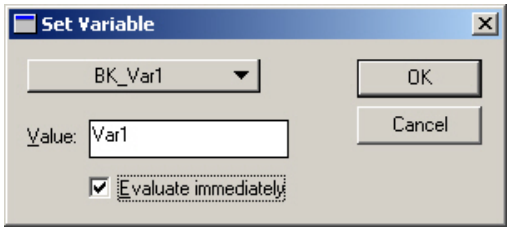



Untitled - Notepad
File Edit Format View Help
Sensitivity, cAntibiotics, 500 - 1000
-- Set variable: cPerDiem = 1000
Sensitivity, cAntibiotics, 500 - 1000
-- Set variable: cPerDiem = 1500
Sensitivity, cAntibiotics, 500 - 1000
-- Set variable: cPerDiem = 2000
Sensitivity, cAntibiotics, 500 - 1000
-- Set variable: cPerDiem = 2500

▷ After making changes, copy the list again, switch back to TreeAge, and paste the list (clearing it first, if desired).

Setting root variable definitions

When specifying a formula definition in the Set Variable dialog, rather than a numeric value, consider using the **Evaluate immediately** option. In many cases, it can be left checked.



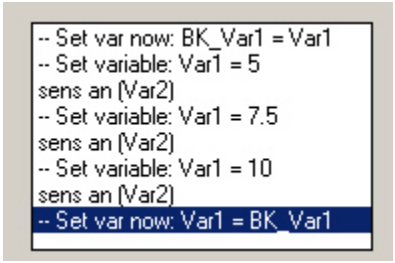
If Evaluate immediately is selected, when the Set Variable step in the sequence is reached, TreeAge Pro will attempt to calculate the formula at the root node and then use this calculated value to create a *numeric* definition of the variable in question (before proceeding to the next step in the sequence). However, TreeAge Pro will be able to create a numeric definition only if all variables used in the formula are defined at the root node.

If a formula is specified in the Set Variable dialog and either (A) the Evaluate immediately option is *not* checked, or (B) Evaluate immediately is checked but definitions are not found at the root node for all variables used in the formula, TreeAge Pro will *not* create a numeric definition of the variable in question. Instead, a text definition of the variable will be created at the root node using the specified formula.

Leave the Evaluate immediately option unchecked if a formula definition specified in an analysis sequence must reference variables that are defined to the right of the root node.

The Evaluate immediately setting will have no effect if the definition in question is numeric.

In the list of sequence events displayed in the Sequence Stored Analyses dialog, variable assignments which do not use the Evaluate immediately option are labeled Set variable: *VarName* = *Value*, while those which do use that option are labeled Set var now: *VarName* = *Formula*.


-- Set var now: BK_Var1 = Var1
-- Set variable: Var1 = 5
sens an (Var2)
-- Set variable: Var1 = 7.5
sens an (Var2)
-- Set variable: Var1 = 10
sens an (Var2)
-- Set var now: Var1 = BK_Var1

Notes:

- The Set Variable dialog can also be used to assign a random value to a variable between analyses, using the Dist() function described in Chapter 16. A sequence can be created that includes a Set Variable command like pCancer = Dist(n; 1). Immediate evaluation of this expression between analyses will result in a random numeric definition of the variable.
- To automate resetting a variable to a default numeric value after a sequence, use the following trick: use an extra variable to store the pre-sequence value. This technique cannot be used to retain formulas.

Stored analyses in a run-time custom interface

This section assumes familiarity with the Run-time version of TreeAge Pro and the related Custom Interface feature, described in an Appendix.

The Maintain Analyses dialog box has some features which may be particularly useful when you are working on a Run-time custom interface. The Run-time user of your tree can be presented with the same list of analyses that appears in the Run Analysis dialog box.

Assign a description and comment for each analysis to be included in the Run-time interface. The long comment is displayed in the main Run Analysis dialog for Run-time users. The summary which TreeAge Pro generates to describe the analysis parameters in the Maintain Analyses dialog is not shown to Run-time users; you may wish to include some of that information in your comment.

You can use the Move Up and Move Down buttons in the Maintain Analyses dialog to change the displayed order of the stored analyses.

To see how the user's Run Analysis dialog box will look, pull down the Options menu, and select Show Custom Interface.

CHAPTER 20

LINKING WITH EXCEL AND OTHER APPLICATIONS

This chapter covers dynamic linking, a powerful tool for integrating a decision tree and an Excel spreadsheet.

Users of the Excel module also have the option of creating a fully-linked influence diagram from a spreadsheet; see Chapters 27 and 29.

In this chapter:

- Dynamic linking with Excel 258
- Calculating payoffs using dynamic links: an example 258
- Using Bilink() to create non-dynamic links 263
- Using DDE links 264

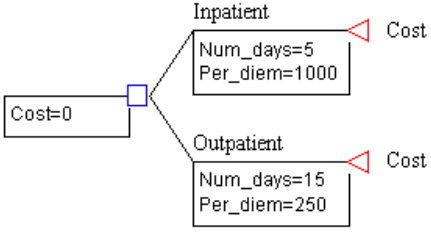
Dynamic linking with an Excel spreadsheet

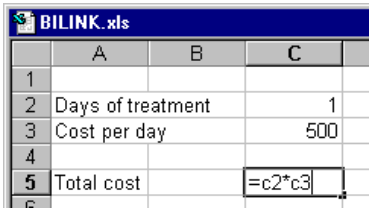
If you have developed a complex set of cost or utility calculations in a spreadsheet, rather than trying to recreate the formula in a decision tree, you can create dynamic links between the tree and the spreadsheet. With dynamic links, TreeAge Pro inputs a set of variables into corresponding spreadsheet cells, recalculates the spreadsheet, and retrieves results from output cells.

TreeAge Pro's implementation of dynamic links is similar to the bi-directional links feature found in the DATA software, but is easier to use, calculates faster, and is highly reliable. Dynamic links work during any type of analysis (e.g. roll back, sensitivity analysis, and Monte Carlo simulation).

To illustrate the basic workings of dynamic links, consider the trivial example tree and spreadsheet pictured below.

Instead of assigning the payoff formula "Num_days*Per_diem" at the terminal nodes, the formula has been set up in a spreadsheet. The variables that are inputs into the payoff calculation, Num_days and Per_diem, are given different definitions at each terminal node. These definitions will be passed to the spreadsheet, which will then recalculate the payoff and pass it back to the Cost payoff variable.




BILINK.xls

	A	B	C
1			
2	Days of treatment		1
3	Cost per day		500
4			
5	Total cost		=c2*c3

Calculating payoffs using dynamic links: an example

Before proceeding, either recreate the simple tree and spreadsheet shown below, or open the copies of the two files – Bilink.tre and Bilink.xls — from TreeAge Pro's Tutorial Examples subdirectory. If you recreate the files, make sure the spreadsheet is saved to disk before continuing.

> ***Excel Module Add-in:***
>
> Users of the TreeAge Pro Excel module have two additional, easier options for setting up dynamic links. Dynamically-linked influence diagrams can be automatically imported from a spreadsheet. Or, all of the steps required to set up dynamic links with a tree can be performed using the TreeAge menu in Excel. See Chapter 27 for more details.

Once you have created the tree and spreadsheet, there are three steps to setting up the dynamic link between them:

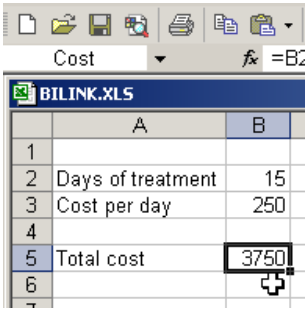
- link the tree to an *output* cell (or range of cells) in the spreadsheet
- assign tree variable *inputs* to cells in the spreadsheet
- define a payoff variable or other parameter in the tree to use the spreadsheet output

Naming a spreadsheet cell

Before TreeAge Pro can connect to any input or output cells in the spreadsheet, these cells must be *named*. The copy of the Bilink.xls spreadsheet included in the Tutorial Examples subdirectory already has names assigned to the output cell (Cost) and the two input cells (Per_diem and Num_days).

► **To assign a name to a cell in Excel:**

▷ Select the cell (or range of cells), click in the Name box (shown at left), type a one word name (like a TreeAge variable name), and press the ENTER key to save the name.



If a cell (or range of cells) already has a name defined, it will appear in the Name box when the cell (or range of cells) is selected.

> ***Tips***:
>
> The other way to assign names to cells in the Excel spreadsheet is by selecting the cell (or range of cells) and choosing Insert > Name > Define…. The dialog displayed is used to manage existing names in the spreadsheet.
>
> When setting up dynamic links, it is helpful to give a spreadsheet cell a name that is similar to the corresponding variable's names (or vice versa, if you set up the spreadsheet before creating the tree).

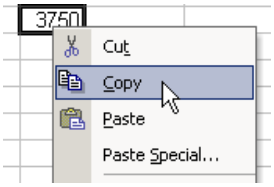
Linking to a spreadsheet output cell

Once an output cell has been named, a link to it can be pasted into the tree.

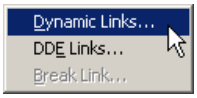
► **To paste a dynamic link output from Excel into the tree:**

⇨ Make sure the spreadsheet has been saved to disk.

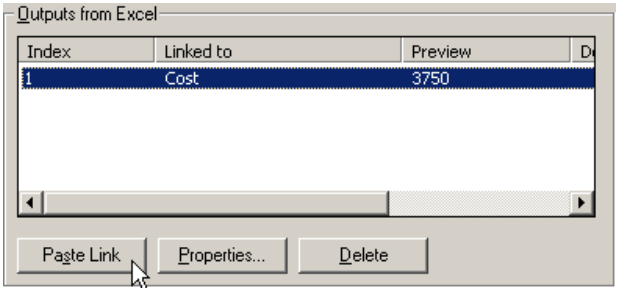
⇨ In the spreadsheet, select the output cell (Cost, in the example) and choose **Edit** > Copy.



⇨ Switch to TreeAge Pro, and choose **Edit** > Links > Dynamic Links…. Click the **Paste Link** button in the Outputs from Excel section.



The link will be displayed in the list of outputs from Excel. Any number of outputs can be pasted into the tree, so each is assigned a numeric index. As is explained below, this integer index will be used in the Bilink() function to reference the appropriate output cell's value in tree calculations. Along with the output link index, the cell's name and current value are displayed.

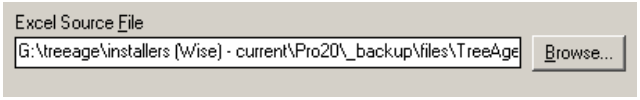


Pasting the first dynamic output link will automatically fill in the required Excel Source File item, at the bottom of the Dynamic Links dialog. All outputs and inputs must be to cells in this one file; the cells can be located on any of its worksheets, however.

The source file can also be selected manually. For example, if the path or name of the spreadsheet changes, all the link outputs and inputs can be pointed to the new spreadsheet by changing the source filename.

► **To change the name or path of the source spreadsheet:**

▷ Open the Dynamic Links dialog. Under the Excel Source File section, either edit the name manually or click the **Browse…** button and search for the file.

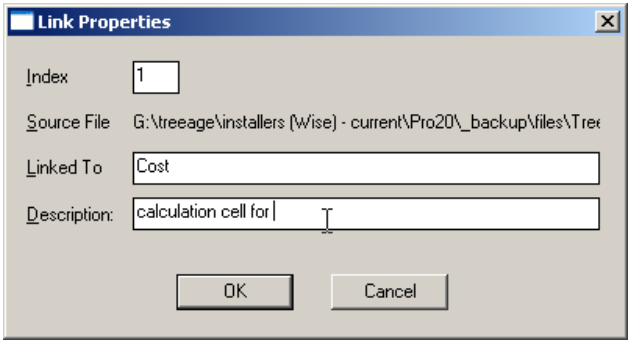


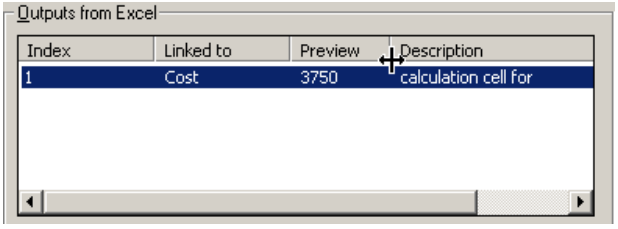
If, instead of changing the source file, you need to change which cell or sheet a particular link refers to, you can either delete the link and recreate it, or update the properties of the existing link.

► **To update a link output cell's name (or sheet):**

▷ Open the Dynamic Links dialog, select a link output, and click the **Properties…** button.

A description entered in the link properties will appear in the Dynamic Links dialog's list of Excel outputs.





Connecting tree variables to input cells (optional)

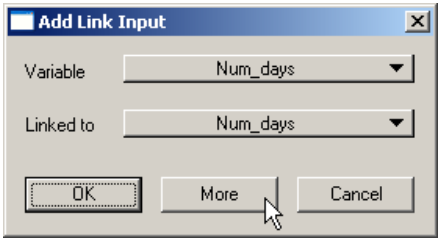
The next step, connecting variables to input cells in the spreadsheet, is what makes a link *dynamic*. If you skip this step (which may be desirable in some cases), the output cell's value would not change during calculations.

In our example, however, we need the Cost output cell to return different values for each terminal node. To accomplish this, we will connect the Num_days and Per_diem variables to the two cells that are inputs to the Cost output cell. Each terminal node will have different definitions of these input variables, and will therefore receive a different calculated value from the Cost output cell.

► **To input variables into named input cells in Excel:**

⇨ In TreeAge Pro's Dynamic Links dialog, under Inputs to Excel, click the **Add…** button.

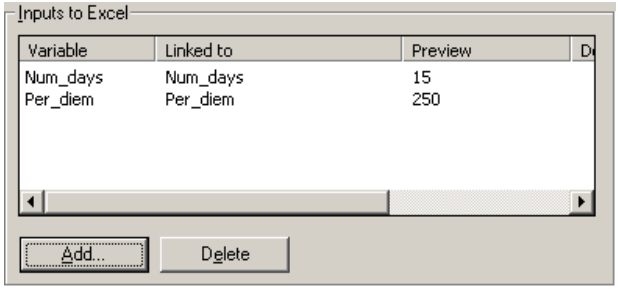
⇨ Select a tree variable and the corresponding input cell. In this case, choose Num_days for both. Click the More button to store this pair and set up another.



⇨ For the second pair, select Per_diem for both the variable and cell, then click OK to return to the Dynamic Links dialog.

The list in the Inputs to Excel section will show each linked variable and input cell pair, along with the cell's current value.

To change which variable a cell is paired with, delete the variable's existing pair and recreate the desired pairing.



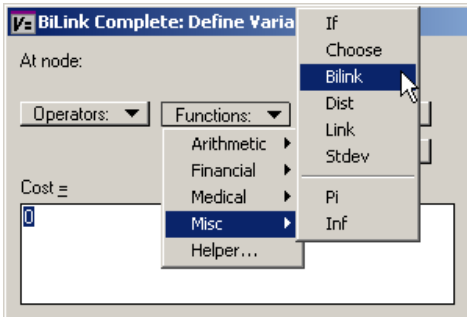
Using the cell output in the tree: the Bilink() function

Finally, the link output(s) must be referenced in the tree using TreeAge Pro's Bilink() function. As described in the above section on outputs from Excel, this function takes at least one integer argument — the index of the output item.

The Bilink(*n*) reference can be used directly within a payoff, or it can be assigned to a tree variable.

► **To use the output cell value in a tree calculation:**

⇨ The Bilink tree already has the variable Cost assigned to each payoff. Now, select the root node and define Cost equal to `Bilink(1)`.

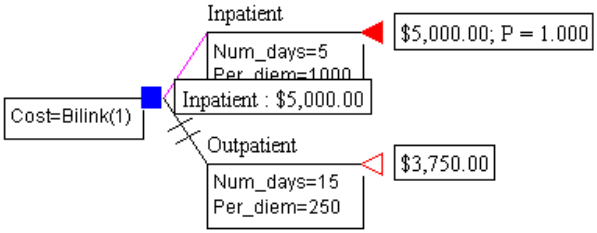


Tip:

If your model uses more than just a few links, in some cases it might make the model clearer if the BilinkName("*range_name*") function is used, rather than the regular Bilink(*n*) function. The BilinkName() function is a new function in TreeAge Pro 2006 (so it is **not** backward compatible with v2005 or earlier). See Chapter 21 for additional details.

Calculation using dynamic links

Now, roll back the tree to verify that different values have been recalculated for each terminal node using the spreadsheet formula.



Once the dynamic link is created, all calculations will utilize the linkage, not just roll back.

Once the dynamic link is set up, even if the spreadsheet is closed when you try to calculate the tree, TreeAge Pro can start Excel and open the spreadsheet automatically. If the a named, linked cell is moved within the spreadsheet the links will not be affected, so as long as the linked cells' *names* are not changed or lost.

Using a single link to a range of cells

If a link is created to a named range of cells, the Bilink() function's index argument is followed by two more integer arguments — the row and column of the desired cell within the range. For example:

Bilink(1; 10; 2)

will retrieve the value from row 10, column 2 of link #1's region of cells.

Using Bilink() to create non-dynamic links

As mentioned above, if a dynamic link is set up without specifying any variables to input to Excel, then the link is not really dynamic. Use of TreeAge Pro's Bilink() function in this way is, in fact, the best way to create *static* links to cell values in an Excel spreadsheet.

If you have a tree created in an earlier version of TreeAge which uses the Sub() function for static, DDE linking (see below) or bi-directional linking, these links will still work in TreeAge Pro. If all of your links are to a single Excel spreadsheet, there are numerous benefits to converting existing links to the new Bilink() function — for both dynamic *and* static linking. The Bilink() function will provide a more stable and flexible means of linking to Excel.

If your tree is linked to other trees, to programs other than Excel, or to multiple Excel spreadsheets, you can continue to use the DDE links functionality described below.

Tip:

When building a tree in TreeAge Pro, the ability to link to cells in a spreadsheet offers a powerful and flexible means of viewing and modifying values in the tree. However, a complex tree making use of links will not calculate as rapidly as a model which does not use links. Also, analyzing a tree making use of the Bilink() function requires that the linked spreadsheet always be present.

Using DDE links

This section covers the use of DDE for exchanging calculated values between a tree and a document in another application, such as an Excel spreadsheet. The use of DDE links to statically exchange calculated values between trees within TreeAge Pro is covered in Chapter 21.

DDE basics

Setting up a DDE link to or from a tree is similar to using the Copy and Paste menu commands for transferring text over the clipboard. The value of a DDE link, however, is updated automatically if both trees are open.

There are a few terms you should be familiar with before proceeding. In DDE, a document or application which makes information available is called a *server*, and a document or application which receives information over a DDE link is called a *client*. TreeAge Pro can act as either client or server, or both.

There are several ways to exchange values via DDE:

- Trees can reference individual cell values, or a range of cells, from a spreadsheet. These values will be updated as determined by the external application. These client-side links are created using the **Edit >** Links > DDE… dialog. They are accessed in formulas in a tree using the Link() function (replaces the Sub() function in previous versions).
- Nodes in a tree can export calculated values for use in external applications. Other documents can link to the expected value (cost and effectiveness separately, in cost-effectiveness models), path probability, and/or standard deviation as calculated at a specified node in the tree. There links are created using the **Edit >** Copy Special command.
- Calculated node values can also be shared between TreeAge Pro models. Trees linked in this way are said to be *nested*. Nested trees are discussed in Chapter 21.

DDE linking to values in other applications

A client-side link can be set up to connect to the value of an item in any DDE-compliant source document.. Typically, this is either a single cell, or a range of cells, in a spreadsheet. There are two steps in creating a client-side link: 1) copying the item in the source document; and 2) pasting the link into the tree.

► **To capture values stored in another document via DDE:**

- ▷ Open the source document – for instance, an Excel spreadsheet. Make sure the file has been saved.
- ▷ Select the item to link to – for instance, a cell, or range of cells, in the spreadsheet. Choose Copy from the program's Edit menu. (Note that some applications have a Copy Link or Copy Special menu item that is more appropriate.)
- ▷ Switch to the tree.

Once the link source has been copied, it can be pasted into TreeAge Pro, using the DDE Links dialog, shown below. The DDE Links dialog can also be used to modify or delete links.

► **To create a DDE link using the Links dialog:**

- ▷ Select **Edit >** Links > DDE….
- ▷ In the DDE Links dialog, press the **Paste Link** button, and note the index of the newly created link. (If the Paste Link button is unavailable after copying a source item, refer to the section on link requirements, below.)
- ▷ Close the dialog, and use the Link(*n*) function to reference the link value in a tree payoff or other expression

In place of a numeric value, any expression which evaluates to a valid integer index can be used in a Link() function. As with the Bilink() function described earlier in this chapter, DDE links to multi-cell ranges can reference a row and column in the selected range. Variable expressions can be used for the row and column parameters of the Link() function.

Client-side link requirements

These guidelines should be followed when using client-side links:

Recalculation – In order for TreeAge Pro to update the value of a DDE link when the source document (e.g., the spreadsheet) is modified, both the tree and the source document must be open, and a tree calculation using the link carried out.

Worksheet order – If there are multiple worksheets in a source spreadsheet, the linked cells should be on the first worksheet (i.e., the left-most tab); see related tip, at left.

Cell names – If you are linking to a named cell in a spreadsheet, TreeAge Pro will link to the name rather than to the cell position. It is advisable to use named cells so that the link will be maintained even if modifications to the spreadsheet cause the cell location to change.

Cell formatting – Linked-to spreadsheet cells should not use currency, or other text formats. TreeAge Pro may correctly create the initial link to a cell with improper formatting; the link will not update correctly when changes are made, though, causing incorrect tree calculations.

As with dynamic links, it is important to name a cell or cell range *before* copying it to create a DDE link. Linking to a named range means that the cell can move around within the worksheet, or the range can increase or decrease in size, without breaking the link.

The Links and Link Properties dialogs

The Links dialog displays the list of client-side links that have been created in the active tree, with the following information for each link:

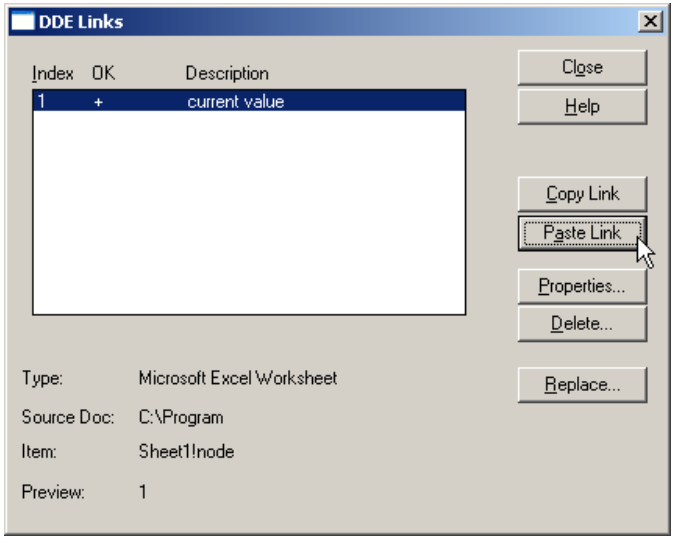
Index: The unique index of the link is used in the Sub() function. When the Sub() function is encountered in a formula, TreeAge Pro uses the most recent value(s) for the link with the specified index.

OK: A "+" will appear in this column in the dialog if the link is currently open. This will occur when the server document is open and TreeAge Pro is able to establish a DDE connection with it. TreeAge Pro will not automatically open DDE link source documents.

Bi-Link: A "Y" will appear in this column when the link is marked as initiating old-style bi-directional links.

Description: This optional field can be used for your own annotation purposes.

The index numbers are initially assigned by TreeAge Pro, but can be changed using the Link Properties dialog. To edit a link's properties, select it from the list and click the Properties button. In the Link Properties dialog, you can change the index integer, source document path, description, and other properties of individual link items.



Changing the location of the client tree will not break the link, as long as the server document is still accessible. However, if the location or name of the server document, worksheet, or cell reference has been changed, the link will be broken, and you must update the link using the Link Properties dialog.

If you need to change many links' paths at the same time, use the Replace Directories dialog. With the Replace Directories dialog, you can quickly update the path property of some or all of the links in the active tree. TreeAge Pro will find all occurrences of the search text in the selected source documents' paths, substituting the new path string you specify. This can be particularly useful if you create multiple links to a spreadsheet, and then have to move the spreadsheet to a different directory, drive, or computer.

► **To replace path strings in many links at once:**

▷ In the Links dialog, select all the links with a common path element to update, and then click on the Replace… button.

▷ Enter a path string to search for, such as "C:\MyDocs", and a replacement string, such as "C:\Windows\Desktop".

▷ If you wish to search only in selected links, check the Replace only within selected items option. If it is not checked, all link items will be searched and updated.

▷ Press ENTER or click OK to update the links.

Note that changing the location of the client tree will not affect the link, as long as the server document is still accessible.

The Copy Link button in the Links dialog box enables you to duplicate one or more client-side DDE links from one tree to another. In the Links dialog of the original tree, select the links that you would like to pass to another tree and click the Copy Link button. Then, open the Links dialog box in the other tree and choose Paste Link.

If bi-directional links were created in an earlier version, the Bi-Directional Links… button at the bottom of the Links dialog opens the Bi-Directional Links dialog.

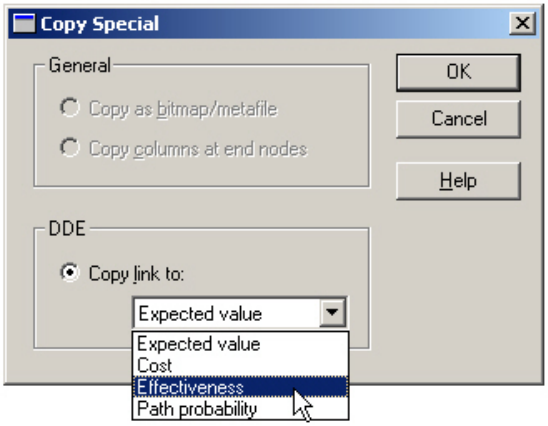
Using TreeAge Pro as a DDE server

Using DDE, TreeAge Pro can send calculated values from a tree to any DDE client document, including spreadsheets and other trees (see the section on Nested trees in Chapter 21). Each node can export via DDE its expected value, path probability, and standard deviation.

► **To use a calculated node value in another document:**

▷ Select a single node whose value you would like to use.

▷ Choose **Edit >** Copy Special.



▷ In the Copy Special dialog box, choose the appropriate radio button for the value you wish to export. Then, close the Copy Special dialog.

▷ The DDE link information is now on the clipboard. In the client document (which may or may not be a TreeAge Pro document) choose Paste, Paste Link, or Paste Special.

> ***Tip:***
> Users of the TreeAge Pro Excel module can use the TreeCalc() worksheet function to dynamically set a cell's formula to a node's expected value.

Each application handles its own client-side DDE links differently; consult the application's documentation for details.

All links to calculated node values will be updated whenever you roll back the tree. In addition, particular calculated values are updated and sent to the client when the corresponding Analysis menu item and node are selected. For instance, if you have copied a link to the path probability of a node Disease, then the link will be updated both upon roll back and when you choose Analysis > Path Probability with that node selected.

When using DDE, both documents must be open for updating to work predictably. If the client document is closed when you roll back the tree, it will not receive these newly calculated values.

If you are unable to arrange all linked cells on a single, top-most worksheet, then you should ensure that both the spreadsheet and tree are open whenever you make changes to values or formulas in the spreadsheet. Linked cells *not* in the top worksheet will only be updated in the tree when the spreadsheet is recalculated while the tree is open.

Chapter 21

Tools for Large and Complex Trees

This chapter focuses on a number of features in TreeAge Pro 2007 that can be indispensable aids to both building and reviewing complex trees.

In this chapter:

- Working with very large trees 272
- The tree explorer view and properties grid 272
- Tree outlines and the Node Outline window 276
- Cloning subtrees 282
- User-defined Python functions/variables 288
- Node(), Tree(), and other special functions 290
- DDE linking trees 298

Working with very large trees

Many of the modeling exercises and examples in thie manual are based on simple, small trees. The trees required in your own projects may be much more complex, perhaps including hundreds of variables, thousands of nodes, scores of distributions, etc. (Which is not to suggest that a bigger model is a better model, generally.)

Review

Previous chapters detailed some of TreeAge Pro's many productivity features designed to make it easier to work with large trees:

- copying and pasting subtrees (Chapter 9)
- collapsing/hiding subtrees (Chapter 11)
- using the Zoom feature to get an overview of the tree (Chapter 2)
- using preferences that compress the tree (Chapter 11)
- building influence diagrams (Chapter 4)

Additional tools/techniques in this chapter

This chapter focuses on additional productivity features relevant to complex modeling projects:

- using the tree's Explorer view, with its highly compact display, for navigation
- making changes using the node Properties grid
- making changes to a node via its text outline, in the Node Outline window
- cloning subtrees (rather than copying/pasting)
- functions for linking node calculations in one subtree to calculations in other trees/subtrees
- creating user-defined functions with Python
- using special functions to perform specialized tasks automatically during calculations (e.g., run a macro, enable debugging, or export the Global matrix)

The Explorer view and Properties grid

The default layout of the tree window includes: the main tree view in the center (white background); the Explorer view docked at top left (grey background); and the Properties grid docked at bottom left. (Other panes may be docked in the tree or floating too: the Debug pane; Variable Definitions window; Node Outline window; Markov Bindings window; Finder. In addition, the TreeAge Pro application includes a Documents View explorer.)

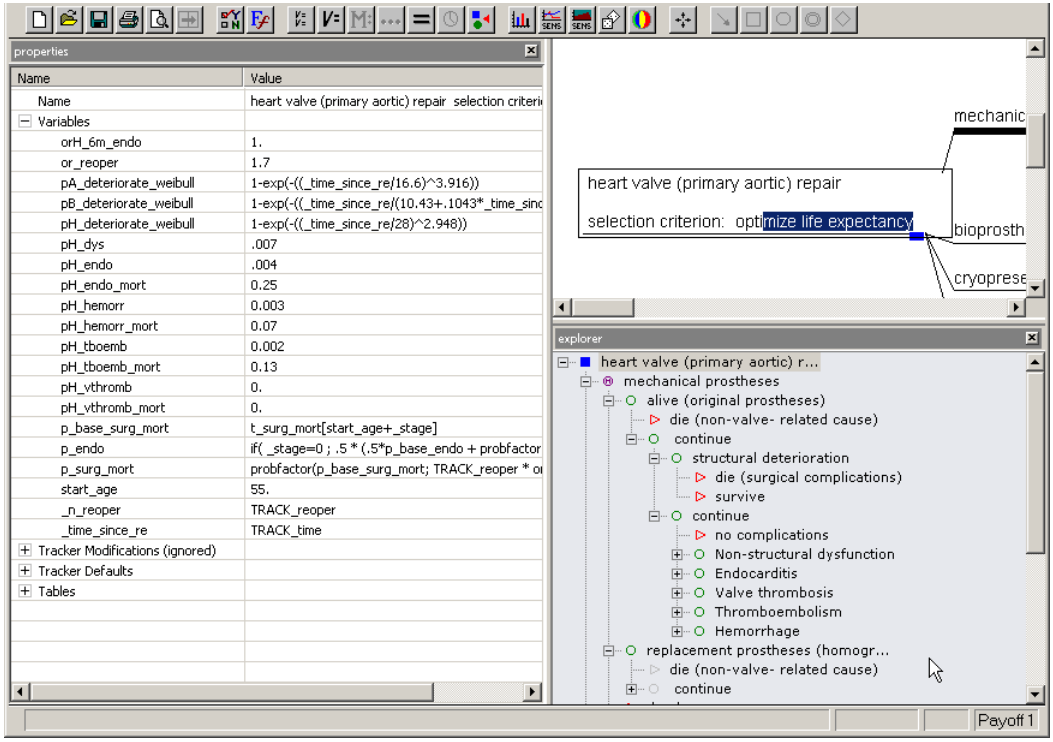
The Explorer view, Properties grid, and other optional tree views each can be separately hidden (from the Display menu) or resized (using the mouse). Each pane can also be docked on different edges of the tree window, or moved to a "floating" position in or outside the TreeAge Pro window.

Docking the Explorer view and other panes

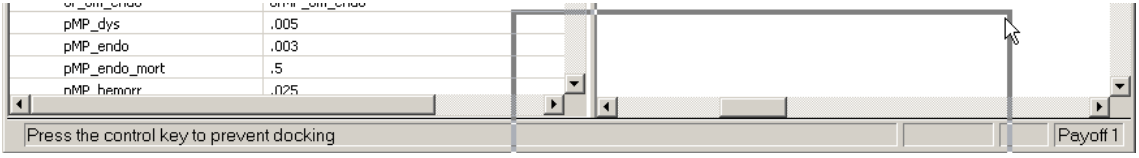
If you have a very large model, you might consider customizing the layout of the tree window in different ways, using the docking capabilities of the tree window panes. To rearrange your workspace for greater efficiency when working in complex models, you might try reducing the space taken up by the tree view and increasing the space used by either the Explorer view or Properties grid. For example, you could try one of the arrangements shown on the next page.

The size and location of extra panes are stored with the tree. And starting with TreeAge Pro 2007, these settings are regular preferences which are stored with the other default settings used when creating new trees.

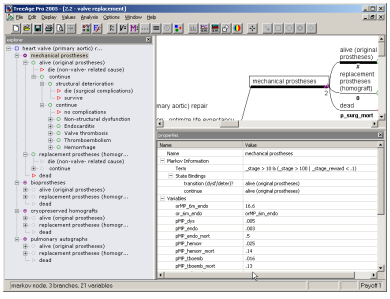
> ***Tip:***
> If both panes are docked to the same edge, their relative positions are determined by the order in which they are docked. To switch their positions, remove both panes to a different edge, and then move them back again. The first pane to be docked on the left or right edge will be on top; the first docked on the top or bottom edge will be on the left.

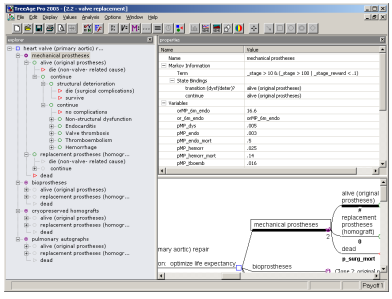


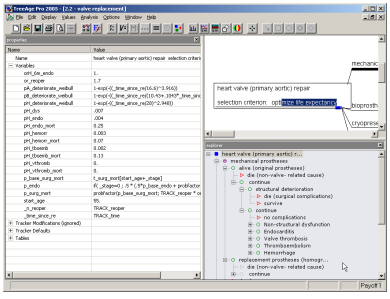
In the upper screenshot, the Explorer view has been removed from the left edge of the tree window and docked on the bottom, simply by clicking on the title bar and dragging it down until it docks.



Try different arrangements of docked (or floating) panes to see what works best for a particular tree.







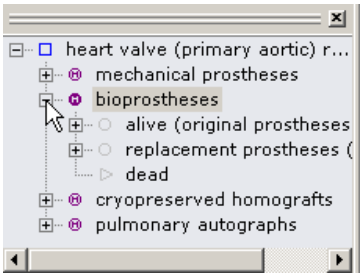
Explorer view and Properties grid details

Most navigation in a tree can be done via the Explorer view. The Explorer view also has a number of special capabilities.

► **To expand or collapse a subtree in the Explorer view:**

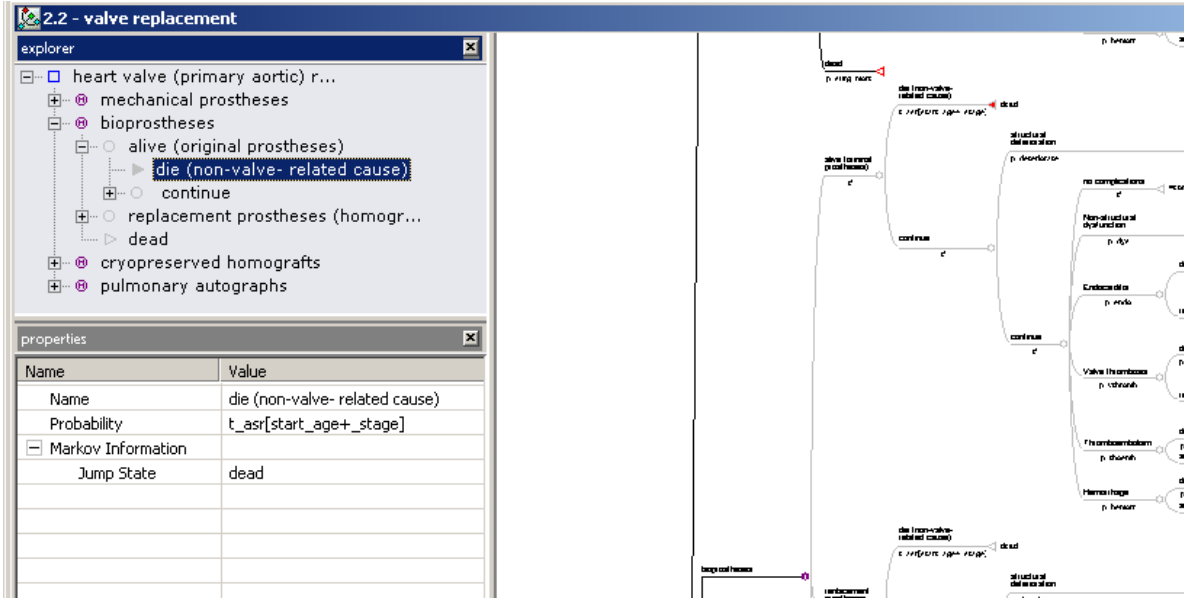
▷ Click on the plus (+) sign to the left of a node symbol to display its hidden branches.

▷ Click on a minus (-) sign to the left of a node symbol to hide its branches (and the rest of the subtree).



A node without a plus or minus sign to the left has no branches/subtree.

Cloned subtrees are shown in gray in the Explorer view, but can be selected there.



Chapter 9 showed how to duplicate or move subtrees in the main tree view using the Select/Cut/Copy/Paste Subtree commands. It is also possible to accomplish these tasks in the Explorer view, without using the menu commands.

► **To duplicate a subtree in the Explorer view:**

▷ Press and hold down the CONTROL key, then click and drag a node with branches to another node.

► **To move a subtree in the Explorer view:**

▷ Click and drag a node with branches to another node.

Many text and value editing tasks illustrated in the tutorial using menus and dialogs can also be done in the Properties grid. Throughout the manual, a small Properties grid icon is used to highlight node editing tasks that can be done with the Properties grid.

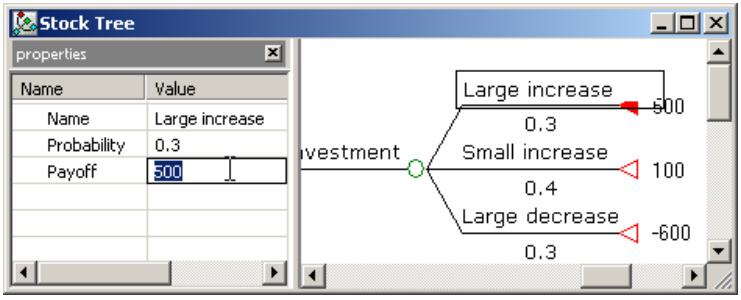
The Properties grid provides editing capabilities for much of the information entered at nodes, including label text, probabilities, payoffs, variable definitions, Markov information, and more. In addition to editable information, the grid also displays some read-only information.

Similar to the Explorer view, the Properties grid displays information in a compact format. You can expand and collapse sections of the Properties grid that have grouped information (such as multiple active payoff sets).

► **To edit values and text in the Properties grid:**

▷ Select the target node, using either the mouse or the keyboard to navigate in the main tree window (or the Explorer view).

▷ If necessary, expand a section of the grid using the plus/minus controls, as needed.



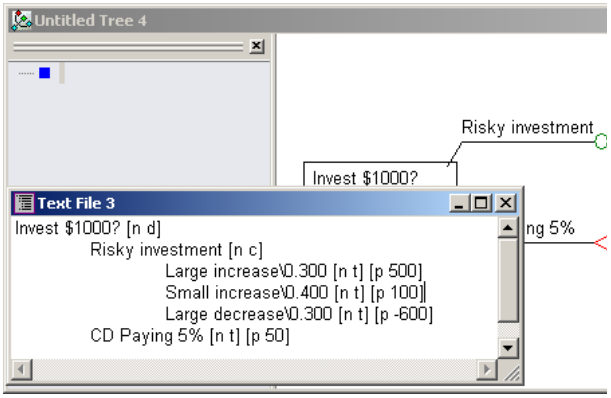
▷ Click on the appropriate row, below the Value column. If the contents are editable, the cursor will change to a text caret, and the active box will be highlighted.

Tree outlines and the Node Outline window

A tree outline is a simple text representation of a decision tree which can be converted into a TreeAge Pro tree file. The text outline can be created in a spreadsheet or word processing program, saved as a text document, opened in TreeAge Pro, and then converted into a tree. It is also possible to type a tree outline in a new, blank text file created in TreeAge Pro using the **File** > New command.

► **To convert an open outline text document into a tree:**

▷ Choose **File** > Convert to Tree.


Untitled Tree 4
Risky investment
Invest $1000?
Text File 3
Invest $1000? [n d]
Risky investment [n c]
Large increase\0.300 [n t] [p 500]
Small increase\0.400 [n t] [p 100]
Large decrease\0.300 [n t] [p -600]
CD Paying 5% [n t] [p 50]
ng 5%

General details on the outline format syntax are provided later in this chapter.

When converting an entire tree to outline format, some information, such as distributions, links, and variable descriptions are not included in the outline. Users of the Excel module can instead use the TreeAge add-in to import/export distributions, links, and other features not supported by outlines. The Excel/COM module's macro/scripting functionality could also be used to create macros/scripts that automate the elements of the import/export process. See Chapter 30 for more on the *TreeAgeProLib*.

The Node Outline window

Starting with TreeAge Pro 2007, the same outline syntax used to build trees can also be used to make quick changes to single nodes within existing trees (without converting the entire model back and forth).

Clicking on the Node Outline toolbar button (or pressing Ctrl+F2, or choosing **Display >** Show Node Outline Window) will display a text pane showing the text outline of the currently selected node. The same commands hide the pane.

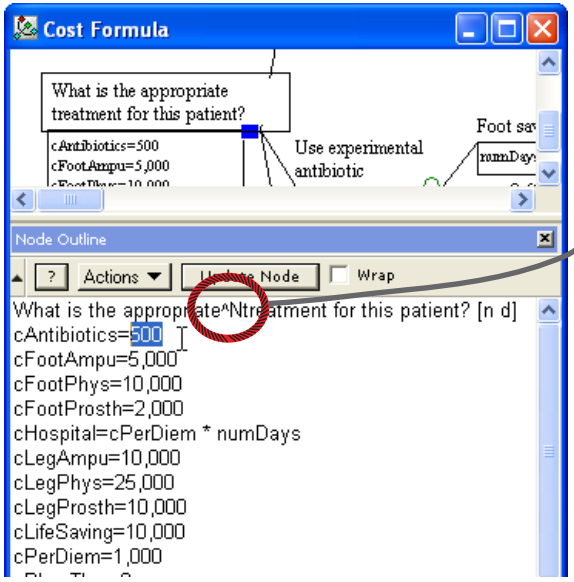


With the Node Outline window displayed, each time you select a node, its outline is generated and displayed in the window. You can then make changes to the selected node.

► **To change a node using the node outline:**

▷ Change text in the window (for example, modify the node type or add/edit payoffs, probabilities, clones, Markov information, variables/trackers, etc.).

▷ Press the **Update Node** button to delete the selected node and in its place create a new node using the modified node outline.



Notes on using the Node Outline window

- To copy/paste text in the node outline, right-click in the text pane.
- Note the use of the "^N" key combination in example below, in order to represent a carriage return in a node label in the outline. Please refer to the following pages for other key syntax.
- It is recommended that you save the model before making a complex node outline change. Most changes can be reversed using the Undo command, but in a few cases information may not be recovered by undo.
- The syntax for the node outline is the same as the tree outline, described in this chapter.
- Works with a single selected node.
- Node properties not shown in the outline format (e.g., fonts) will be lost if you update via the outline. Properties related to calculation and reporting (name, variables, values, clone info, notes/comments, etc.) are handled.
- The window is a dockable, hideable pane, and its display preferences are stored with the tree.
- It is possible to create a new (blank) tree, and use the Node Outline window to enter/update with a full tree outline. Additional, indented nodes/subtrees in the outline will be added to the root node.

Tree outline basics

In an outline, indentation is used to represent the generational levels found in tree documents. For example, a tree fragment can be specified in outline form by:

```
Root node
   Child One
   Child Two
```

The use of indents indicates that Child One and Child Two are children of Root Node. In a text document, indentation is accomplished with the TAB key; using the space bar will not work. To indent in a spreadsheet, move to the next column. The number of tabs found before a node name indicates the generation to which the node belongs. The root node of the tree must be on the first line with no leading TAB characters.

Each node must be on a line by itself. The entire contents of a node must be on a single line. If your text editor wraps the text of a long line, you must ensure that no RETURN character occurs within the wrapped text. A multi-line node label may be created on a single line by inserting the character sequence "^N" where a line break should be generated.

An example outline

The following outline demonstrates some of the important concepts in outline creation.

```
Decide^Nnow.[n d]<-Root node
   Chance Subtree[n c] [ci 1] [cn P1]
      High Payoff\.8[n t] [p PayoffOne]
      Low Payoff\#[n t] [p PayoffTwo]
   Clone-copy[n c][v V1=14\V2=2][cc 1]
PayoffOne=60 - V1
PayoffTwo=50 - V2
V1=10.9
V2=1
```

Try copying this into a text file, opening it in TreeAge Pro, and converting to a tree.

Node outline details

Most of the internal details of the node, such as its variables, are specified in the outline by using brackets. The general form of the bracket construct is:

[key *value*]

where "key" is one of the character codes listed in the following table, and "value" is dependent on the key. The allowed "key" characters are:

Key	Meaning
n	Node type
v	Variables
p	Payoffs (or Markov jump-to state name)
t	Markov node termination condition
r	Markov state rewards
u	Markov tunnels
j	Markov transition rewards
cc	Clone copy
ci	Clone master index
cn	Clone master name
m	Node comment (this branch)

The key characters are not case sensitive. There may be one or more spaces between the key and the associated value field. Bracketed keys may appear in any order on a node line.

Branch name and branch probability

The first text on each line in the outline is the node name. Then, the probability for a branch can be specified by following the node name with a backslash ("\") and then the probability value or formula, as shown in this example:

```
Root node [n d]
   Child One [n c]
      Possibility One\0.4 [n t]
      Possibility Two\0.6 [n t]
   Child Two [n t]
```

Node types

To specify the type of node directly, use [n *x*], where *x* is one of the following characters:

Value	Node Type
d	Decision
b	Decision, optimal path switched
c	Chance
m	Markov
l	Logic
a	Label
t	Terminal

The node-type characters are not case sensitive. If the type is not specified explicitly, or incorrectly, it will be determined in the following manner:

- If a node is not already a chance or logic node, and its children have probabilities, it becomes a chance node.
- Otherwise, the node becomes an undefined node.

Payoffs

To specify payoffs for a terminal node, use [p *x*], where *x* is of the form:

PayoffVal1\PayoffVal2\PayoffVal3\...

The payoffs are exactly as specified in the Enter Payoff dialog in the tree window. Note that any payoffs omitted will be left blank, as in:

PayoffVal1\\PayoffVal3

In this case, payoffs #2 and #4 through 9 will be left empty.

This same rule is used for Markov rewards.

Definitions of variables

You may specify variables for any node in a tree or node outline by using [v *x*], where *x* is of the form:

Var1=Value\Var2=Value

Specify one or more variables by listing the variable's name, followed by "=" and then its value. Multiple variables are separated by the backslash character.

In a node outline (or the root node of a tree outline), however, there is an easier approach to creating lists of variable definitions. "Default" definitions can be listed at the bottom of the outline. Here is an example tree outline:

```
Root node [n d]
   Child One [n c] [v Cost=1000]
   Child Two [n c] [v Cost=1500]
Profit=Revenue-Cost
Revenue=Units*PricePer
```

In this case, the fourth and fifth lines are not interpreted as nodes, since the root node cannot have siblings. You may list several variables, each on its own line. Note that, unlike variables defined on node lines, default definitions are not surrounded by brackets.

Use the same approach in a node outline (in the Node Outline window).

To add a definition comment, simply append a "#" after the definition followed by the comment:

[v Var1=Value# comment here\Var2=Value]

User-defined function definitions

The format for a user-defined Python function is:

```
(F) def userFuncName(param=0):
        return param**2; # square it
```

User-defined Python functions are covered in detail in this chapter.

Cloned subtrees

Clones may be created in the outline in two steps. First, you must create a master, which has a name and an index. Then, you create a copy by giving a node the index of the master to which you want it attached. To specify a master node, use the following two fields on the node's line:

[ci *index*] [cn *name*]

Here *index* is a positive integer index you assign to the master, which is used only within the outline. It may be any positive integer. You name the clone by using *name*, which will be used to attach clones in the tree window only.

To attach a clone to another node, add the following field to the copy node:

[cc *index*]

where *index* is the index number you assigned to the master node.

Note boxes

A note box can follow the text of any node. It must be the last text on the line, and may not include carriage returns (use ^N instead). You indicate the start of a note using the "<-" combination:

Branch name\0.3<-This is a comment.

Upon converting the outline into a tree, a note box will be created, and bound to the associated node.

Tracker variables

Definitions of tracker variables, described in the Healthcare module documentation (Chapter 36), are specified in the outline using the prefix "{T}" before the name of the variable:

```
{T} countEvents=countEvents+1
```

(Curly braces are also used to indicate a table lookup; see below.)

Table lookups

In an outline, curly braces {} are used to indicate a table lookup (for example, in probabilities), because square brackets are reserved for other purposes.

When converting to a tree, TreeAge Pro will automatically handle the conversion between square brackets and curly braces.

Markov transitions and transition bindings

You may specify the jump-to state for a transition node by using the key/value pair usually used for terminal node payoffs:

[p *state*]

where *state* is the jump-to state's name.

To create a Markov transition binding (see Ch. 35) in the outline, create a variable definition at a node in the Markov process, but use an invalid name for the variable/binding, as in:

[v *binding name*=*state name*]

where *binding name* has one or more spaces (or other characters that are not acceptable characters in regular variable names). The binding name (prefixed with an equal sign) can then be used as the jump-to state at a transition node somewhere to the right, as follows:

[p =*binding name*]

Markov termination condition

You may specify the termination condition for a Markov node by adding a key/value pair:

```
[t cond]
```

where *cond* is the termination condition (such as "... [t _stage=5]"). Enter multiple sets of conditions using the payoff syntax (backslash separators).

Markov state and transition rewards

To specify the rewards for a Markov state, add a key/value pair:

```
[r x]
```

where *x* should be in the form of payoffs:

```
Init 1\Incr 1\Final 1\Init 2\Incr 2\...
```

Only one *init\incr\final* group is required. To skip a set of rewards, do not enter any text between the backslashes.

Transition rewards can be added at nodes (e.g., *jump* nodes) to the right of a Markov state by adding the key/value pair:

```
[j x]
```

where *x* is of the form:

```
Rew1\Rew2\...
```

Only one reward is required. To skip a reward set, do not enter any text between the backslashes.

Markov tunnel states

Markov tunnel states are created by adding a key/value pair to the Markov state:

```
[u x]
```

where *x* is the number of temporary states, exactly as entered in the Markov State Information dialog.

Cloning subtrees

In addition to being able to duplicate existing subtrees using the Copy/Paste Subtree commands, it is also possible to create *clones* of subtrees. The basic difference between attaching clones and pasting copies is that a copied/pasted subtree can be edited; these copies do not automatically update when changes are made to the original. Clone copies, on the other hand, are linked dynamically to the original, *master* subtree, and are not directly editable. A clone copy always automatically takes its structure and other contents from its clone master (unless the relationship is explicitly broken).

Notes:

- Multiple clone masters can be used in a tree, and can be nested.
- Variables (and state bindings) **used** in a clone master can calculate differently in clone copies if the variables are **defined** outside the master/clone.
- Cloning only works within a single tree.
- A clone master copied as part of a larger subtree will *not* be pasted as a clone master (or as a clone copy). A clone copy, when copied as part of a larger subtree, will be pasted as a regular subtree, not a clone copy.

Creating clone masters and attaching copies

The first step in the cloning process is to select a clone *master* subtree for the purpose of being replicated at other locations in the tree. The clone master does not have to be a finished structure; all that is required is a single branch. The master subtree can continue to be updated after copies of it have been attached to other nodes in the tree.

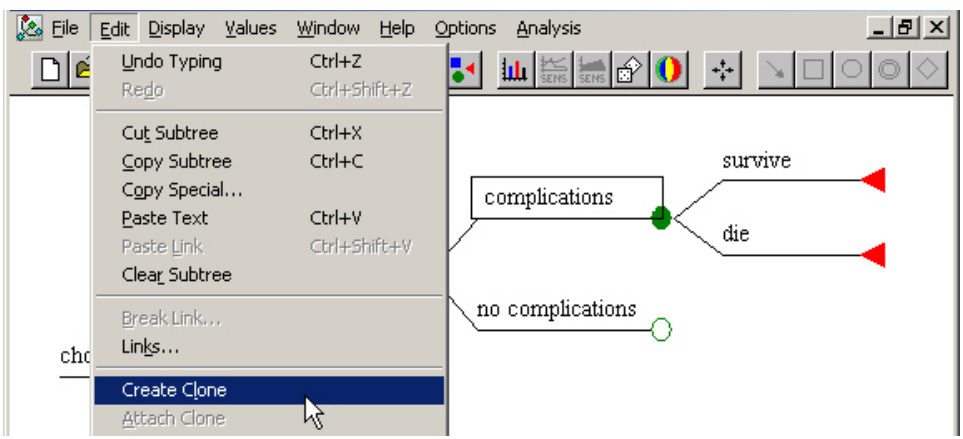
Each subtree designated a clone master can be identified by a heavy bar beneath the branch leading to its root node. Like a copied subtree, the clone master does not actually include its root node.

► **To create a clone master:**

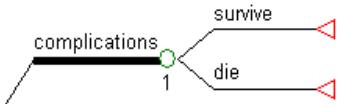
▷ Identify the master subtree and select it. To select a subtree, either hold down the CONTROL key while clicking on the root node of the subtree, or select its root node and choose **Options >** Select Subtree (see Chapter 8 for more information on working with subtrees).

▷ Choose **Edit >** Create Clone.

▷ Provide a short, descriptive name to identify this clone master. If the preference to display clone copies is turned off (see below), this name will appear in place of the structure, next to the root node of each attached clone copy.



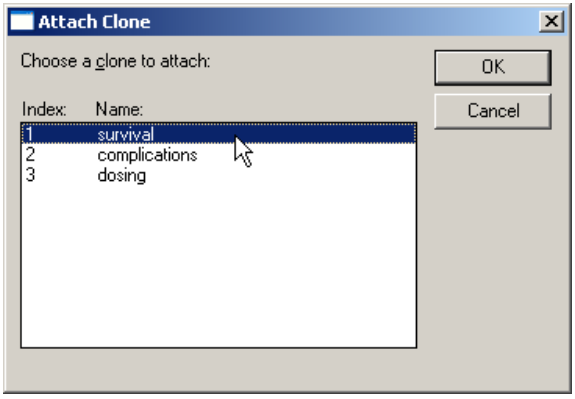
Clone masters are also assigned a numeric index, which appears next to the clone master node.



Once the clone master has been created, the next step is to attach clone copies at appropriate node(s).

Clone copies can only be attached to nodes which have no branches. The nodes where you attach a clone copy should have the same node type as the root node of the master subtree. Attaching a clone copy to a node will *not* automatically change the type of the node to match the clone master's root node. This change can be made manually, if necessary, either before or after attaching the clone copy.

Copies of a clone master can be attached to more than one node at a time, if multiple nodes are selected.



► **To attach clone copies:**

▷ Select one or more appropriate nodes (i.e., a node with no descendants).

▷ Choose **Edit >** Attach Clone.

▷ If you have only a single clone master in the tree, it will be attached to the selected node(s) automatically. Otherwise, you must select the name of the appropriate clone master from a list, and click OK.

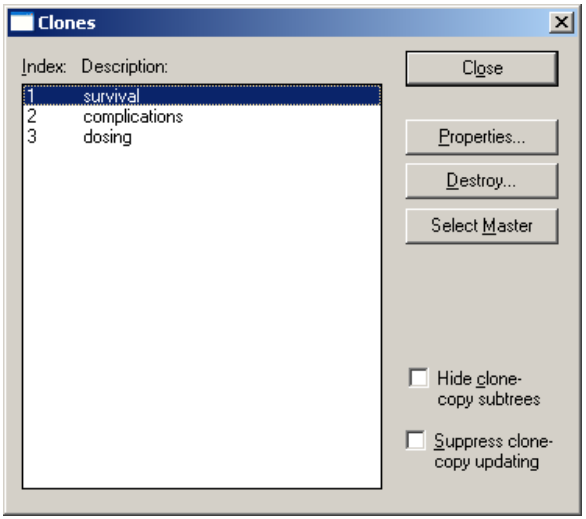
It is possible to change a clone master's index number or name once clone copies of it have been attached, without breaking the linkage.

► **To change a clone master's name or index number:**

▷ Choose **Edit >** Clones…, select the clone to edit, and click the **Properties…** button.

Destroying and detaching clones

If you want to eliminate all clone copies of a particular clone master, you can either destroy the current clone master, and automatically detach its clone copies, or detach only selected clone copies one at a time.



There are two ways to destroy a clone master.

► **To eliminate a clone master in the tree:**

▷ Select the clone master *subtree* (not just the node), for example by CONTROL-clicking on the root node of the clone master.

▷ Choose **Edit >** Destroy Clone.

► **To eliminate a clone master using the Clones dialog:**

▷ Choose **Edit >** Clones to open the Clones dialog.

▷ From the list, select the appropriate clone master, and click on the Destroy button.

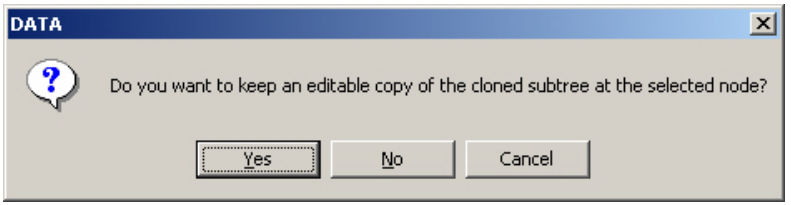
Destroying a clone master will "un-publish" the subtree. The subtree which was formerly the clone master will remain in the tree; all of its clone copies will be removed.

If you have attached clone copies based on a master subtree which you are going to destroy, but you wish to replace the clone copies with regular (i.e., unlinked and editable) copies, you can detach each clone copy.

► **To detach clone copies:**

▷ Select one or more node where a clone copy is attached, and choose **Edit >** Detach Clone.

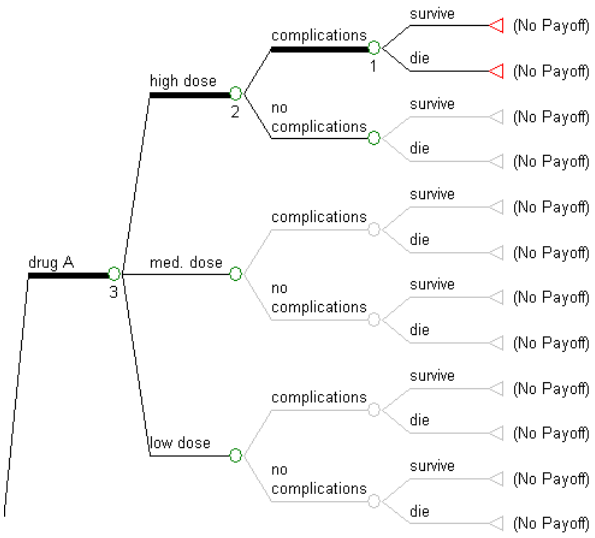
You will be given the opportunity to leave a regular copy of the master subtree when a clone copy is detached. If you need to change which clone is attached at a particular node, detach the current clone copy and specify that you do not want to keep an editable copy of the cloned subtree at the selected node.



Nesting clones

An important aspect of clones is the ability to nest clone masters. In other words, a single subtree may be comprised of multiple, independent clone masters, and also include various clone copies, as well.

The tree shown below has three, nested clone master subtrees. Clone masters 2 and 3 include nested clone masters and copies.



You cannot attach a copy of a master subtree to itself.

Using variables within clones

Calculations in clone copies are performed as if a full copy of the master subtree existed at the location of the clone copy. This is true even if you have turned off the display of clone copies in the tree (see below).

Since clone copies are identical to the clone master, variables must be used in the clone master if you want its copies to have different probabilities and other values. The probability, reward, and payoff expressions in the clone master and its copies will all use the same variables. However, each will be able to use different values for these variables, based on how the variables are defined outside the clones. By defining variables used in the clones differently at the respective root nodes of the clone master and each cloned subtree, appropriate values can be calculated within the clone copies.

The file Cloning Example provides a relatively simple example of using variables with clones. See Chapters 12–14 for more on variables.

Hiding clone copies

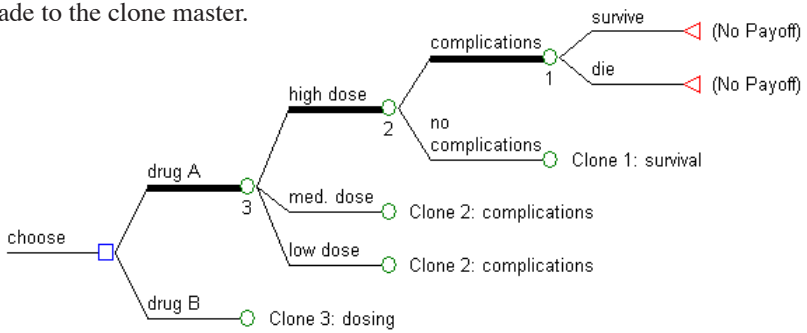
If clone copies are visible in the tree, their nodes and branch lines are drawn in gray, in order to distinguish them from editable subtrees. One of the advantages of using clones is that the replicated, clone copy structures in your model can be hidden from view, thereby reducing the overall size of the displayed tree, often quite dramatically. The Tree Display preferences include an option to hide clone copies.

► **To hide clone copies:**

▷ Choose **Edit >** Preferences, and go to the Tree Display category in the Preferences dialog.

▷ Check the box labeled Hide clone copies, and click OK to return to the tree.

When clone copies are hidden, just the name of the clone master is displayed to the right of the node where the copy is attached. Hiding clone copies does not affect calculations in any way. Hidden clone copies will be updated when changes are made to the clone master.



A common by-product of reducing tree size by hiding clone copies is enhanced clarity. The essential features of the replicated subtree may be understood by examining the clone master, which is the only instance of the subtree that is displayed. In addition, the cloning linkages within the model, which might otherwise be missed, are clearly visible, as each clone copy indicates the master to which it is linked.

Model-building strategy for very large clones

Even while using the Suppress Clone Updating feature described on the next page, working on very large trees composed of many thousands of cloned nodes still requires efficient model-building strategy. If you are building such a model, it is generally an efficient strategy to complete your clone masters **before** attaching any clone copies – and, thus, before clone updating suppression is useful/needed.

If you have to make structural or value changes to the clone masters later, once clones copies have been attached, then you can make good use of the suppress clone updates feature.

Suppressing clone copy updating

TreeAge Pro includes a feature for speeding up work on clone-heavy trees. The Display menu (as well as the Edit > Clones… dialog) includes a command to temporarily turn off the updating of clone copies. If clone suppression is turned on, existing clone copy nodes will be hidden and changes that you make to clone masters will not immediately cascade through clone copies.

Normally, each change you make to a clone master forces a regeneration of the clone copies. As you add more clones to your model, especially nested clones, the time required for clone updating grows larger. Suppressing the updating of clone copies will significantly speed up the process of modifying a model with many clone copies.

► **To clone copy updating on or off in a tree:**

▷ Choose **Display >** Suppress Clone Updating.

Turning this setting on also hides existing clone copies, if not already hidden using the tree preference described earlier in this chapter.

Performing an analysis will turn off the suppression of updates and cause clone copies to be regenerated. It will be necessary to turn clone copy updating back on manually if you want to create new clone masters or attach clone copies. The same menu command used to turn off clone copy updating is used to turn it back on and regenerate the clone copies.

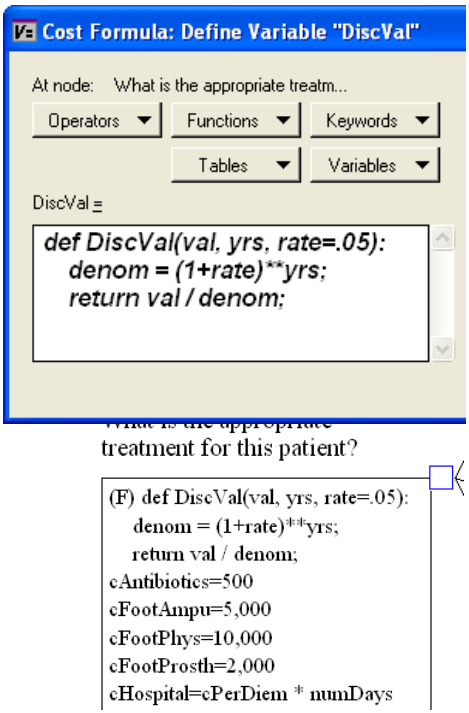
Tips:

- Turning on the Suppress Clone Updating feature works fairly quickly. Turning it off, on the other hand, will result in a delay while the updated clone copies are regenerated. An efficient approach is to wait until you have a batch of changes to make to the clone masters before turning on the suppression of clone updates.
- The commands used to create new clone masters, and attach or detach clone copies, are available while clone updating is suppressed. Note that detaching a clone copy while clone updating is suppressed will only give you the option of leaving a copy of the first cloned subtree, not any nested clones.
- A tree can be saved with clone updating suppressed, for example if you have to close the file before you finish making changes to the clone masters. When reopened, the tree will still be in the suppressed state.
- Simply hiding clone copies using the Tree Display preferences is not the same as suppressing clone updating.
- It is possible to turn on both the hide clone copies and suppress clone updating settings while a tree is being opened, before it is displayed. Select the File menu, but before clicking the Open command, press and hold down the CONTROL + SHIFT + ALT keys until the tree is finished opening.
- To avoid using up memory in very large cloned trees (*not* Markov models), a setting in the Clones dialog can be used to dynamically unsuppress clone copies as they are needed during analysis, and then resuppress them.

User-defined Python functions/variables

TreeAge includes a variety of built-in functions (see Chapter 14). However, if there is no built-in function that does what your model requires, you can write a simple user-defined function in Python script, and embed it in a variable in your tree.

Previously, using Python script in TreeAge Pro was not easy. Starting with version 2007, however, Python functions can simply be written within a definition of a regular variable.



User-defined function syntax

A user-defined function in TreeAge is contained within a variable definition. For example, a variable called DiscVal could be added to the tree. Then DiscVal could be defined at a node, perhaps the root node, with the following function entered as its definition:

```
def DiscVal(val,yrs,rate=.05):
    denom = (1+rate)**yrs;
    return val/denom;
```

The function declaration is the first line in the definition:

```
def DiscVal(val,yrs,rate=.05):
```

After the Python keyword `def`, repeat the variable's name using the same case; ***Python is case sensitive***. After the name, parentheses are required; inside the parentheses, an optional argument list can be entered, separated by commas. Starting on the right, arguments can be given default values (making them optional). Finally, the declaration ends with a "`:`" (colon) and a carriage return.

Lines following the declaration must use tabs or spaces to express indent levels, which is how Python interprets blocks of code. Statement lines are ended/separated with a "`;`" (semicolon). Note the use of # to mark comment lines in the examples of the next page. The end of the function is implicit.

Calling the user-defined function

The point of having the user-defined function, of course, is to be able to call it from a payoff, probability, or other calculation in the tree. If your user-defined function specifies required arguments, then you must call it using a User( ) function:

```
payoff = User("DiscVal"; costX; _stage; rate)
```

Use the original variable name's case to refer to the variable/function everywhere, as Python is case sensitive.

If no arguments are required for a particular user-defined function, define it without arguments:

```
def pi_squared( ):
     return 3.1415**2;
```

and simply call the function from a payoff, probability, or other tree calculation using the function's variable name (without parentheses):

```
rate = pi_squared
```

Using the "treeage" object in Python script

Although some tree variables and values will be passed to the user-defined functions via the list of arguments, it is also possible for the functions to access tree variables and functions directly:

```
def pi_treeage():
   # treeage object's eval method
   return treeage.eval( "PI( )" );
```

The `treeage.eval()` call will occur in the context of the current tree calculation (e.g., a node whose probability uses the `rate` variable). Therefore, any variables that are referenced by the eval() call must be defined at or to the left of that node.

Another function supported by the treeage object is addTableRows, illustrated here:

```
   ... function body ...
   # "addTableRows()" returns count
   #   of rows added to TreeAge
   #   "myTable" from list of lists
   rows = treeage.addTableRows(listX,
"myTable", true);#  true = clear table
```

Why TreeAge Pro uses embedded Python

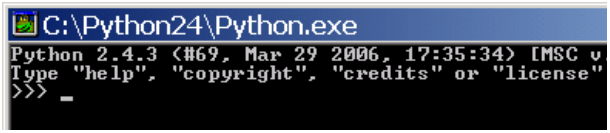
Python is an easy-to-use object-oriented programming and scripting language, including standard elements like looping, type conversion, etc.:

```
for i in range(1 to 20):
   s = str(i);
   ct=ct+treeage.eval("ctEvt"+s);
```

Your user-defined functions can also import classes and functions from Python's powerful set of built-in modules, as well as from the vast array of custom, open source Python modules.

```
... function body ...
from math import log10;
return math.log10( val );
```

To support user-defined functions, the TreeAge Pro installation includes the core Python modules only (in a library called PythonXX.dll, in the TreeAge Pro application directory). If you wish to test Python scripts outside of a TreeAge model, install the standard Python distribution. This will, for example, enable you to use a Python command-line interpreter.



To download Python (free, open source) and learn about available modules, visit:

http://docs.python.org

For a good reference guide to the Python language and its capabilities, try:

Python in a Nutshell, 2nd Ed. (O'Reilly, 2006)

Notes on TreeAge Pro's Python interpreter

Calls to the Python equivalent of built-in TreeAge Pro functions will be slower than simply calling the built-in TreeAge Pro function, due to overhead in reading text strings, etc.

Use of the Python functions requires the Python 2.4 library, which is a default option in the TreeAge Pro installer.

For more information, see:

http://www.treeage.com/support/Python.html

Node(), Tree(), and other special functions

TreeAge Pro includes a variety of special functions which can be useful in complex models. Brief introductions to two of these functions – Node() and Tree() – are provided first.

String-arguments and concatenation

A variety of functions use text/string function arguments (rather than typical arguments which are either numeric or variables/expressions with a numeric value in the tree). Most string arguments can be constructed using concatenation; in other words, text and calculated values can be combined into a new string.

For example, the filename in the Tree( ) function could be concatenated (with a variable called "pick", in this case):

```
Tree("myTree_"+pick+".tre";"Node(1;0)")
```

Similarly, the worksheet name in the Command("Excel"; "ExportGlobalMatrix";...) function could be concatenated with a variable:

```
Command(…;"sheet_"+_trial+"_";…)
```

The concatenation can include simple numeric expressions, variables, or other expressions that should be evaluated before being added to the string.

Using the Node() function

The Node() function can act as an internal linking function, within a particular tree, allowing nodes in different parts of a tree to communicate. Some possible uses for the Node() function include:

- As described in Chapter 7, the Node() function can be used to access calculated expected values in non-active payoff sets.
- The Node() function offers a means of reusing a subtree at different locations in a tree — this is similar to the way cloning is used, but with greater flexibility. For example, as described in Chapter 35, the Node() function can be used to nest a Markov model within another Markov model, or to enable expected value calculations in trees that previously were limited to analyses using Monte Carlo microsimulation trials.

Chapter 7 illustrated the simplest form of the function, using the syntax:

Node(*attribute*)

where attribute is an integer from 1 – 9 referring to a non-active payoff set.

In most cases, the Node() function's arguments are used to pick a node in the tree and a type of calculation to perform there, for example:

Node(*attribute*; *method*; *branch list; …*)

The *branch list* is a series of branch numbers used to pick a node to calculate. At least one branch number must be specified, to select a branch of the root node. To select a branch of that branch, add a second integer to the list (and so on). The numbering of a node's branches starts at 1, for the top branch.

Node() function syntax

The Node function is discussed in Chapters 7, 10, 21, and 35. It offers powerful functionality for building complex models and customizing TreeAge Pro calculations. It is recommended that the Node function be tested in simple models before being used in complex models.

Node(Attribute)

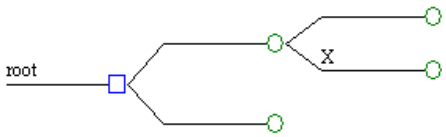
Returns the stored expected value of the current node, including from an extra payoff. Attribute is an integer corresponding to payoff 1 through 9. It can be used, for example, to report extra payoffs in custom roll back columns (see Chapter 10).

Node(Attribute; 0; LIST)

Returns the expected value from a node whose path is defined by LIST. The Node( ) function can be called from within another Node( ) function call.

Attribute:
Use 0 to skip the calculation and return 0; in Cost-Effectiveness calculations, use -1 to perform CE calculations and return the new cost calculation; then use -2 to return the stored effectiveness calculation. (Use -11 to return a stored cost calculation, or -12 to recalculate the node and return a new effectiveness calculation. In non-CE models, use any number other than 0 for attribute.

LIST:
Use a series of branch numbers to select a node (e.g., ";1;2" selects node X in the tree shown below).



Node(Attribute; trials; LIST)

Runs a series of n=*trials* microsimulation trials at the node whose path is defined by LIST. Returns the average of the trials results, normally. The Node( ) function can be called from within another Node( ) function call, or any other expression (including the Tree( ) function, discussed below).

It is important to note some differences in behavior in Node( ) function trials, as compared to regular Monte Carlo microsimulation trials. The Node( ) function does *not* reset tracker variables to their default values before each trial (allowing trackers to be used to track information across the group of trials). Distributions do not resample per trial, by default. To force distribution sampling, use tracker variables set equal to the DistForce(*n*) function, for example.

Attribute and LIST:
See above.

Trials:
To return a statistic other than the mean value from the set of trials, add a decimal after the *trials* value, and then add a number corresponding to the appropriate statistic:

mean = 1	median = 2	mode = 3	standard dev = 4
10% = 5	90% = 6	2.5% = 7	97.5% = 8

Using the Tree() function

TreeAge Pro includes a number of functions which take double-quoted text string arguments, rather than just arguments that can be evaluated numerically. Like all functions in TreeAge Pro, the string-type functions return a numeric value.

Text string function arguments can be constructed using concatenation; in other words, text and calculated values can be combined into a new string. For example, the filename in the Tree( ) function could be concatenated (with a variable called "pick", in this case): *Tree("myTree_" + pick + ".tre"; "Node(1;0)")*

It is recommended that these functions be carefully tested in simple models before being used in complex models.

Tree("file"; "calculation")

Opens a hidden copy of the specified tree or package file, and returns the results of the specified calculation. The hidden tree remains open for additional Tree( ) function calls.

"File":
Specify the path and filename of the saved tree (or package) to open for calculation. Or, if the file is in the same directory as the current tree, use just the filename.

"Calculation":
Usually the Node( ) function will be used, in order to calculate a specific node in the hidden tree. However, it is also possible to calculate root node variable names, expressions, and even run TreeAge scripts.

Example usage: utilXYZ = Tree("c:\my docs\projects\xyz.tre"; "Node(1;0)")

The Node( ) function, in this case, performs a simple expected value calculation at the root node.

Notes:

Both .tre and .pkg files can be opened and calculated using the Tree( ) function.

When the function accesses a hidden tree, a temporary linkage is created: the hidden tree's root node becomes the "child" of the calling node. The power of the Tree( ) function is that a hidden tree opened by the calling tree can use variable definitions from the calling tree – either at, or to the left of, the calling node. So, in some cases, a variable referenced in the hidden tree will not have a definition in that tree, but only in the calling tree (the name must be declared in the hidden tree, however).

The function's primary use is for modularization of complex trees. For example, it might offer an alternative to memory-intensive models in which hundreds or thousands of cloned copies of a Markov model are created. For example, a thousand terminal node payoffs could use the Tree( ) function to calculate the same Markov model (rather than attaching clones of the model at every end node). Only one copy of the tree and its nodes would need to be in memory, with each terminal node in turn taking over the hidden tree.

In the case of a model using a moderate amount of memory, however, calculation speed will generally be better if the Tree( ) function is *not* used, but rather cloning or other modeling techniques are used. TreeAge Pro cannot optimize expressions in the hidden tree that reference a variable defined, not in the hidden tree, but in the calling tree. On the other hand, using clones or regular copied nodes can sometimes result in a tree whose memory requirements exceed available physical memory, resulting in a more extreme decline in calculation speed. Building smaller trees linked by the Tree( ) function would be a good solution in this case.

Tree("calculation")

Can be used during an ongoing Tree( ) function evaluation in order to perform the given calculation at the root node of the current hidden tree.

Tree(integer 0)

Closes any open, hidden trees. Use this, for example, if you open a helper tree using the File menu and save changes to it. This will cause the next Tree( ) function call to reload the tree file from disk.

The User() and Python functions

As illustrated earlier in this chapter, TreeAge Pro includes an embedded Python programming language script interpreter, accessible via user-defined functions/variables. To evaluate a user-defined function requiring one or more arguments as input, the User() function is utilized:

User("myFunction"; argument 1 ; ...)

Example usage: (assuming user-defined function StdNorm(mu,sd) and variables: mu_X = 0, sd_X = 1)
var_X = User("StdNorm"; mu_X; sd_X)
= [e.g., return sample from Python's random.normalvariate function]

See the note at the beginning of this section on using string concatenation to build string arguments.

The Python interpreter is also accessible from a set of string-type functions:

Python("expression") If the current Python can successfully evaluate the variable name, expression, or function, the Python() function will return that value. If it cannot evaluate it, an error is returned. The Python() function returns a numeric value.

Example usage: varA = Python("10+1") = 11
varHex = Python("int('0xff', 16)") = 255

PythonFunc("module"; "function"; arguments ; ...)

Example usage: (assuming variables: mu = 0, sd = 1)
costY = PythonFunc("random"; "normalvariate"; mu; sd)
= [random number from Normal(mu,sd)]

PythonGetList("list"; "treeage table")
The PythonGetList() function copies a two-dimensional Python nested list into a TreeAge table. The Python expression can be a literal list (as below), a variable containing a list, or a function returning a list.

Example usage: rows = PythonGetList("[[0,.1,.2],[1,.15,.25]]" ; "myTable") = 2

PythonRun("command"; ...) Returns 0 for failure or 1 for success. Specify a valid Python statement. To run a series of scripts or modules, separate multiple double-quoted paths using semicolons. If Python successfully runs the last specified statement, the PythonRun() function will return a value of 1. If Python encounters an error evaluating your commands, the PythonRun() function returns a 0 value – ***not an error***. To force a TreeAge Pro error, use the result as the denominator of a division operation. To return the value of variables or functions created or modified by the executed commands, use the Python() function.

Example usage: ok = PythonRun("from random import *" ; "seed(1)"; "x=random()") = 1
value = Python("x") = .134364...

PythonRunFile("scriptfile" ; ...) Returns 0 for failure or 1 for success. Specify the full path and filename of a valid Python script text file or module to run; or, specify just the file name if it is stored in the Python default path. To run a series of scripts or modules, separate multiple double-quoted paths using semicolons. (Note that forward slashes can be used instead of backslashes in the path specification.)

Example usage: ok = PythonFile("C:\Python24\lib\TransplantQueue.py")

Using PythonRunFile() to execute a file is equivalent to having Python evaluate the execfile() function:

ok = PythonRun("execfile('C:\Python24\lib\TransplantQueue.py')")

If the script/module executes successfully, use the Python() function to return the value of variables:

queueWait = Python("now()")

Command() function

Like the Script() function, the Command() function can be used to perform a variety of special commands in the tree window, including updating/clearing tables, adding/deleting branches, etc. For example:

Command("TABLES"; "myTableCost"; "REQUERY")	Reruns an existing table's query, if it is linked to a database.
Excel-module only:	
Command("Excel"; "Open"; Workbook)	Syntax for opening a spreadsheet given a path/filename.
Command("Excel"; "Macro"; "myMacro"; param1)	Syntax for running an Excel macro.
Command("Excel"; "GetCellOffset"; cell; rowOffset ; colOffset)	Syntax for getting a cell's value or text .
Command("Excel"; "SetCellOffset"; cell; rowOffset ; colOffset; value)	Syntax for setting a cell's value or text .
Command("Excel";"ExportGlobalMatrix";Workbook;SheetName;CellName;LabelText)	Export the global matrix to a new or existing sheet. See information on Global() in this chapter.
Command("Excel";"ExportGlobalMatrixN";Index;Workbook;SheetName;CellName;LabelText)	Export numbered matrix to a worksheet. See information on GlobalN() in this chapter.

Note that "CellName" does not currently create a named cell, but rather will try to use the existing named cell. The "BookName" argument and subsequent arguments are optional. They can be all skipped, or empty quotes used to skip selected arguments.

Like all string functions, Command() arguments can use concatenation, as in:

```
Command(...; "sheet_" + varX + "_n_" + varY; ...).
```

See the note at the beginning of this section on using string concatenation to build string arguments.

Contact <support@treeage.com> for help using the Command() function, information about undocumented functionality, or if you have a suggestion for a new syntax/command.

Script() function

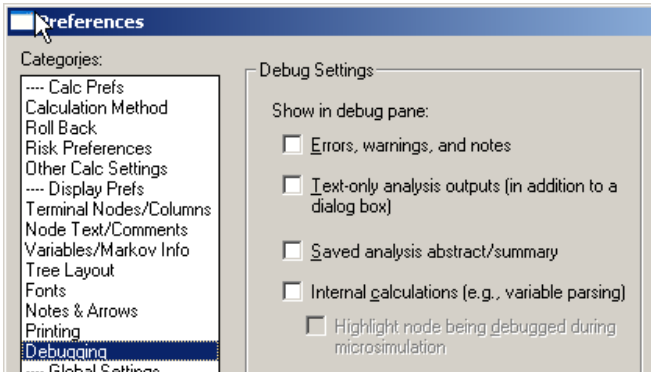
Normally, TreeAgeScript commands (see Appendix F, and the TreeAge Script help file) are entered into a text file, and then the text file is run from the TreeAge Pro File menu. The Script( ) function allows scripts to be run by calculating a variable, instead. One possible application is to run analyses or other commands on hidden trees loaded using the Tree() function.

See the note at the beginning of this section on using string concatenation to build string arguments.

Script("script commands" ; ...)	Runs the specified command(s). The Script() function will return 0 by default or, if any script variables are encountered in the script, it will return the value assigned to the last script variable. To run a series of script commands, either use spaces to separate multiple commands within a single double-quoted text string, or separate multiple double-quoted text strings using semicolons. The former will generally be faster, while the latter will ensure that errors are reported separately for each command. **Example usage:** varA = Script("!SelectRoot() !SelectNode(A) !Markov(1)") >> Attempts to run a cohort analysis at the first branch of the root node, displaying a Markov output window.
Script("script file"; ...)	Run a saved script file. To run a series of script files, separate multiple double-quoted paths using semicolons.

Using the Debug() function and the Debug pane

The Debug pane is an optional part of the tree window that can be used to display/report a variety of textual information, including error messages, text-only analysis output, stored analysis summaries, and detailed internal calculations (e.g., step-by-step evaluation of variables).



If you turn on detailed calculation reporting in the tree's Debugging preferences, each probability, payoff, and other calculation is reported in detail. During Monte Carlo simulations, it reports each random walk and sampled value. See Appendix C for a description of each of the Debug Settings options.

The Debug( ) function can be used to dynamically control Debug pane output during analyses (or to add custom text to the pane). This is of primary interest in complex analyses, when the full debugging output of calculations may be hundreds of thousands of lines (or more) per analysis. Use the Debug( ) function to dynamically turn on and off the flow of calculation outputs to the Debug pane at strategic points in the analysis.

Debug("1") or Debug("0")	Turns on/off detailed calculation debugging.

To report only the calculations of a single probability during an analysis, for example, first turn off the debugging output of Internal calculations in the tree's Debugging preferences. Then, surround the probability by two Debug( ) function calls turning debugging on and then off:

```
probX = Debug( "1" )*0 + prob_of_event + Debug( "0")*0
```

To add custom text to the debugging output, use Debug( ) function calls with a different syntax:

Debug("text")	If detailed calculation debugging is turned on, the function will be shown, as well as the text (concatenated, if necessary).
Debug("text"; "title"; timeline)	Adds a title, and a timeline if timeline is non-zero.

Global() and GlobalN() functions

These advanced functions can be very powerful tools for accomplishing complex calculations and reporting tasks. It is recommended that they be tested in simple models before being used in complex models.

Function	Explanation
Global(i; j; value) GlobalN(n; i; j; value)	(**Note:** In v2007, Global() always refers to the same matrix, unless batches are running on distributed servers. Previously, different threads on a multi-processor computer would use different matrices. GlobalN() for a particular *n* also refers to the same matrix even during a multi-threaded simulation.) **GlobalN( ) has an extra integer parameter *n* >= 1 which picks a matrix to use/create.** The regular Global( ) function uses a separate matrix. Both functions calculate the *value* argument and sets a "cell" in a global matrix/table equal to the result. The function also returns that value. The first cell in the global matrix is at row=1, column=1. Therefore, anywhere *value* is used in an expression, it can instead be "wrapped" in the Global function, in order to save the calculated value to the matrix (for example, for reporting purposes). A value saved to a global matrix can be referenced using the second form of the Global() function (see below). The entire contents of the global matrix can be dynamically saved to a text file using the third form of the Global() function The matrix size is dynamically increased, as necessary (if i and/or j are larger than any previously specified values of i and/or j), up to a maximum of 10e8 cells (e.g., 10000x10000 or 50x2000000). This is the maximum total for all matrices in the tree (e.g., 4x5000x5000). **If i = 0 and j = 0, the calculated value is applied to the entire existing matrix**, allowing it to be dynamically reset during calculations, if necessary. If any argument (not just value) is an expression, it is evaluated at the node being calculated, just as if the expression were not inside the Global() function. The global matrix is deleted when the tree is closed. It is emptied automatically before the next analysis (use the last form of the function, with no arguments, to manually empty the matrix).
Global(i; j) GlobalN(n; i; j)	Retrieves the value stored at row i, column j, in a global matrix/table (e.g., for use in a tree calculation). The first cell in the global matrix is at row=1, column=1. See note on GlobalN( ) above.
Global(value) GlobalN(n; value)	If *value* evaluates to a non-zero number, the current global matrix/table is silently saved to a text file. If *value* evaluates to zero, a text file is not created. If *value* is less than zero, the matrix is reported in the Debug pane as well. The function always returns a value of 1. See note on GlobalN( ) above. If a text file is saved, it is saved in the same directory as the tree (or, if the tree has not been saved, in the root directory of the hard drive). The text file will use the name of the tree followed by the word "globals", an underscore, and the number that *value* evaluates to. For example, a calculation in "c:\trees\mytree.tre" that uses the function "Global(21)" would save the current matrix to a text file called: "c:\trees\mytree_globals_21.txt" NOTE: Refer to the **Command("Excel";"ExportGlobalMatrix";Workbook;SheetName;CellName;LabelText)** function in this chapter for information on exporting the matrix to a spreadsheet instead.
Global() GlobalN(n) GlobalN()	The global matrix (or matrices) is emptied; size is 0. (This occurs automatically at the start of an analysis, or when the tree is closed. At the end of an analysis, however, the matrices are still available via the Global() or GlobalN() functions.)

List/matrix functions

These advanced functions can be used to do matrix operations, for example. It is recommended that they be tested in simple models before being used in complex models. As an alternative, try Python user-defined functions instead.

List(LIST)	Used to contain a parameter list for a multivariate distribution (e.g., Dirichlet). See Chapter 35.
Matrix(cols; LIST)	Stores a list of parameters into a matrix with a specified number of columns; for use as one of the parameters of the MatrixMult() function.
MatrixMult(i; j; m1; m2)	Performs matrix multiplication on two matrices (correlating a 1xN matrix of random numbers using an NxN covariance matrix, for example). Returns a cell from row i, column j, of the resulting matrix: m3(i, j) = sum(m1(i,:)*m2(:,j) The two input matrices, m1 and m2, must conform to required shapes for the matrix multiplication to work: the number of columns in matrix 1 must equal the number of rows in matrix 2: (# columns in m1) = (# values in m2 divided by # columns in m2) Note that the first argument in the Matrix() function is the number of columns (i.e., number of items per row): The resulting matrix has the same number of rows as m1, and the same number of columns as m2. To correlate a list of n values in matrix 1 using a second, n x n variance-covariance matrix, specify n columns for matrix 1 and matrix 2. The resulting matrix of correlated values will have 1 row and n columns.

BilinkName() function

If your model uses more than just a few links (see Chapter 20), it might make the model clearer if the BilinkName("range_name") function is used, rather than the regular Bilink(n) function. The BilinkName() function is one of the new, string-type functions in TreeAge Pro (so it is not backward compatible with v2005 or earlier).

See the note at the beginning of this section on using string concatenation to build string arguments.

BilinkName("cell_name")	Calculates or returns the value of the dynamic link to the named cell/range. See Chapter 20.
BilinkName("cell_name"; row)	For a link to a named range, pick a cell by its row and/or column.
BilinkName("cell_name"; row; col)	

DDE linking trees

For trees which do not require dynamic recalculation of sub-models, it may be possible to use DDE links to connect master and subsidiary models. DDE links can be created to expected values, path probabilities, and standard deviations calculated in a tree.

When not to use DDE linked/nested trees

The use of nested trees has drawbacks when it comes to performing analysis on model parameters, such as sensitivity analysis or Monte Carlo simulation on the master tree. Nested trees are not dynamically linked, which means that the only way to update a link value in the master tree is to first manually recalculate the subsidiary tree. When a parameter changes during sensitivity analysis or simulation on the master tree, the subsidiary tree is not updated or recalculated automatically.

See the descriptions of the Node() and Tree() function, above, for alternatives to nested trees.

Using a subsidiary tree to calculate a probability

You may wish to employ a subsidiary trees to model the probability of an important outcome in the master tree. In this case, you would create the subsidiary tree, probably using only chance nodes, that models one or a series of secondary events whose occurrence would influence the probability of the primary event in the master tree.

To calculate a probability for the primary outcome, assign a payoff of 1 to each endpoint in the subsidiary tree that represents the outcome of interest, and 0 to the others. The *expected value* of the completed subsidiary tree, between 0 and 1, should represent the *probability* of the primary event in the master tree.

After creating the subsidiary tree, simply create a link to its expected value utilizing DDE . With the link information on the clipboard, the dynamic link between trees can be created.

► **To calculate a probability using a subsidiary tree:**

▷ Select the root node of the subsidiary tree.

▷ Choose **Edit >** Copy Special..., and paste the link (using **Edit >** Paste Link) into the probability field of the node in the master.

Rather than referencing the link directly in the probability field, a variable can be created in the master tree to represent the primary outcome's probability. That variable can be then be defined default for the tree using the link reference.

When the master tree is calculated, it will evaluate the branch probability using the last available expected value from the subsidiary tree. Changes made to the subsidiary tree will be reflected in the master tree's calculations through this probability. Both trees must be open when the subsidiary tree is recalculated, in order for the master tree to update the link value.

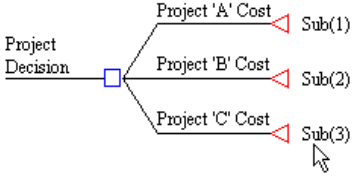
Using a nested tree to calculate a payoff

A subsidiary, linked tree can also model the payoff of a terminal node (or nodes) in the master tree.

► **To calculate a payoff using a subsidiary tree:**

▷ Select the root node of the subsidiary tree.

▷ Copy a link to its expected value by choosing **Edit >** Copy Special..., and paste the link (using **Edit >** Paste Link) into the payoff of the relevant terminal node(s) in the master tree.



Chapter 22

Advanced Chance Node Techniques and Options

This chapter provides information on a number of features which can be used to customize chance nodes in decision trees, some of which are new in TreeAge Pro.

In this chapter:

- Using non-coherent probabilities 300
- Sampling probabilities from a Dirichlet distribution 302
- Using DistKids( ) to create an array of subtrees 304
- Distributing a fixed set of chance node branches 306

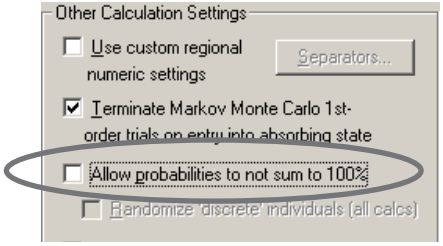
Using "non-coherent" probabilities:

In order to avoid certain calculation errors, TreeAge Pro normally requires the branch probabilities of each chance node to sum to 1.0. Probabilities that meet this requirement are referred to as "coherent." However, there are some situations where it may be useful to relax or remove this restriction. For this reason, TreeAge Pro offers the option of turning off the error checking that normally protects against mistakenly assigning non-coherent probabilities.

You are urged to employ this option cautiously, only after giving careful consideration to the potential hazards of turning off the error checking.

► **To disable errors when using non-coherent probabilities:**

▷ Choose **Edit >** Preferences…, and select Other Calc Settings.


Other Calculation Settings
Use custom regional numeric settings
Separators...
Terminate Markov Monte Carlo 1st-order trials on entry into absorbing state
Allow probabilities to not sum to 100%
Randomize 'discrete' individuals (all calcs)

▷ Check the option labeled Allow probabilities to not sum to 100%. The sub-option, Randomize 'discrete' individuals…, is described below.

As precautions against unintended use, TreeAge Pro will not save this preference as a default setting for new trees, and will prompt you to confirm this preference each time a tree using it is opened.

> ***Note:***
> Non-coherent probabilities are not compatible with microsimulation (first-order trials), but can be used in probabilistic sensitivity analysis using expected value calculations.

Reporting future net present values in the tree

When net present value discounting is used in a tree, the payoff calculation TreeAge Pro reports at a terminal node and uses in expected value calculations to the left normally reflects that scenario's value discounted back to the present time. Therefore, expected values reported at intermediate nodes in the tree are in terms of present value — not value at the time of the intermediate event or decision. Some kinds of analysis, however, such as options valuation, may require calculating net present values at intermediate nodes in the tree in which a scenario's value is discounted back only to that future date, rather than to the present.

In situations where it is appropriate, it is possible to use non-coherent probabilities to perform the discounting, instead of discounting each terminal node payoff. Net present value discounting is generally performed by dividing by *(1+discount_rate)^time*. By turning off probability error checking, this division could instead be performed on each branch probability. If done correctly, the expected values calculated at the root node of the tree will be unchanged from a standard, payoff discounting model. See the TreeAge web site for more information.

As with other uses of probability non-coherence, care must be taken to ensure that hidden probability errors are not unknowingly introduced.

Discrete and dynamically-sized Markov cohorts

Normally, decision trees are used to calculate an expected, or average, value. In budget-oriented or population-based modeling, however, the ultimate goal may be to determine not an average, but an overall cost or benefit. In some cases, this can be accomplished simply by multiplying the expected value by some number (e.g., the number of projects, or the size of a population). If TreeAge Pro's probability coherence requirement is turned off, an equivalent option would be to do the multiplication via tree "probabilities."

Non-coherent probabilities might be used to model a population whose size changes over time. In a Markov model built using the Healthcare module, for example, the starting population can be initialized — sized and distributed among possible health states — prior to the Markov calculation. Then, during Markov calculations, non-coherent probabilities could be used to change the size of the population, modeling entry from other populations (i.e., from uninfected to infected) or internal population growth (births).

When allowing non-coherent probabilities, a sub-option to randomize "discrete" individuals is available that will maintain integer probabilities at subsequent chance nodes — in effect, keeping individuals whole by randomizing them at chance nodes (during any analysis, not just simulation). This might be relevant, for example, where small probability events are critical (i.e., in a vaccination model where continued transmission of a contagious disease requires a "whole" carrier).

See Chapter 35 for more information on this advanced topic.

Sampling probabilities from a multivariate Dirichlet distribution

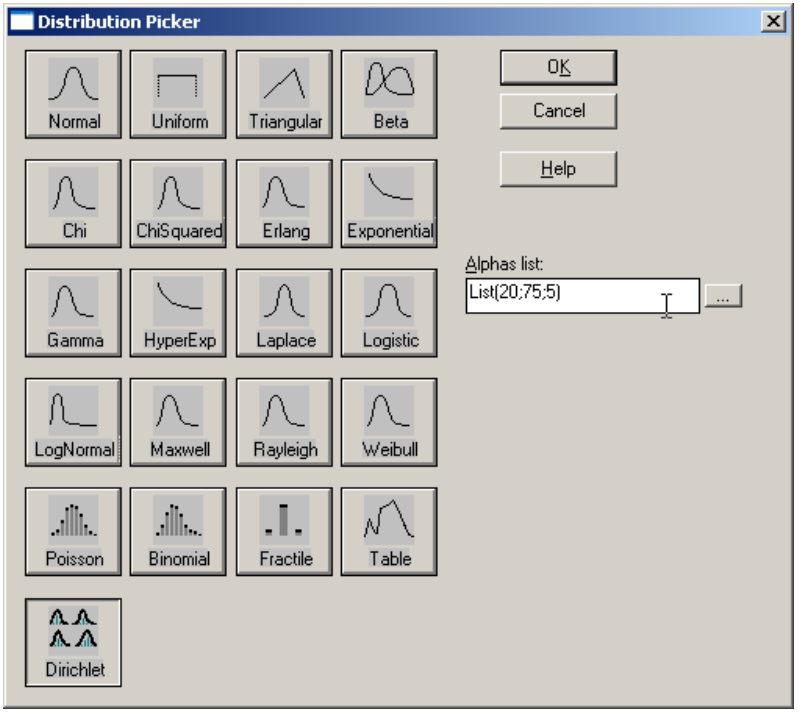
If a chance node has more than two branches, performing a sensitivity analysis or Monte Carlo simulation that changes the values of these probabilities can be problematic.

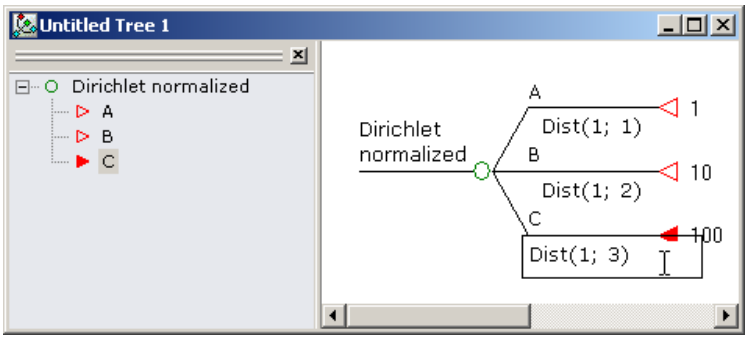
One option is to write expressions that normalize the chance node's probability expressions. For example, let's say the node has three outcomes, A, B, and C. Rather than assigning variables to two probabilities, and using the # remainder calculation in the third, you could do the following: assign three expressions that always sum to 1.0, like `pA/(pA+pB+pC)` and `pB/(pA+pB+pC)` and `pC/(pA+pB+pC)`. No matter what values (>= 0) are assigned to pA, pB, and pC, the three normalized probabilities will always sum to 1.0. (The # remainder could still be used in place of one of these.)

TreeAge Pro offers another solution using a special, multivariate form of the beta probability distribution, called a Dirichlet distribution. This distribution can be used to represent the uncertainty in all of the probabilities of a chance event. During Monte Carlo simulation, the distribution can sample probabilities for each branch, using normalization to ensure that the probabilities always sum to 1.0.

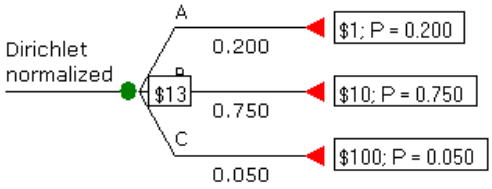
If the distribution is parameterized with a list of three positive *alpha* values, as shown here, TreeAge Pro will samples three independent Gamma[*alpha*, beta=1.0] distributions and normalize these to create a list of three probabilities.

To utilize the generated sample probabilities, the Dist() function described in Chapter 16 is used with a second argument to indicate which branch (i.e., alpha) to use, as shown on the following page.

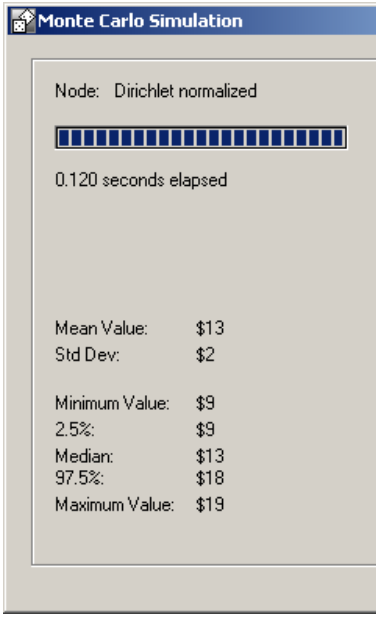




Rolling back the tree shows the mean values of the probabilities, which are simply the normalized alpha parameters.



Performing a simulation in the example tree shows the effect of sampling independent Gamma distribution values, based on the list of alpha parameters, and then normalizing. For each iteration of the simulation, a different set of Gamma random variates is drawn. Each iteration results in a different sum, as well as different ratios of the Gamma random variates to the sum (i.e., the probabilities), but normalization ensures that the resulting probabilities sum to 1.0.



Note:

In TreeAge Pro, simulation text reports and graphs report only the first sampled Dirichlet probability. However, it is possible to use either a tracker variable or TreeAge Pro's Global() matrix function to store and report each sampled probability. For example, to create a table of a branch's sample values, wrap the Dist() function reference in the Global() function, as in:

Global(_sample; branch; Dist(1; branch))

See Appendix D for more details on using the Global() function, including how to dynamically save a text file of the Global matrix.

Using DistKids() to create an array of subtrees

As noted in Chapter 1, a chance node is a structural representation of a probability distribution. In Chapter 16, which deals with Monte Carlo simulation, another representation of a probability distribution was introduced — a sampling distribution, which is not a structural element in the tree, but is instead a numeric input into a probability or payoff/reward calculation.

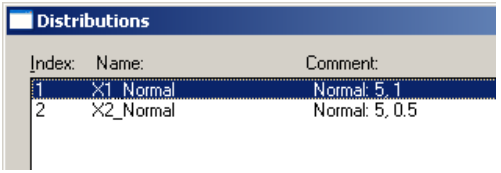
TreeAge Pro offers two methods for combining the specific advantages of a chance node and a sampling distribution. The more flexible, dynamic option involves use of the DistKids() probability function at a chance node, as described in this section. The other option, the Options menu's Distribute Children command, is covered in the next section.

If your model includes a series of decisions, there may be one or more uncertainties that are resolved prior to a decision. In many cases, such an uncertainty might be represented with a chance node preceding the decision node. The problem is that in order to represent an uncertainty's probability distribution with sufficient detail, more than just two or three branches may be necessary (for example, when an important parameter has a *non-linear* effect on calculations; see Chapter 16). Structurally representing even a small number of uncertainties in this fashion can result in a very "bushy" tree.

One solution is to visually represent such uncertainties with a chance node having a single branch, and during any calculation to invisibly expand this chance node into a fuller probability distribution. An appropriate probability distribution must be added to the tree using the Distributions dialog, as it would be for Monte Carlo simulation. The distribution will be referenced twice in the tree: first, in the probability expression of the chance node branch's probability; second, in a payoff or other tree calculation to the right.

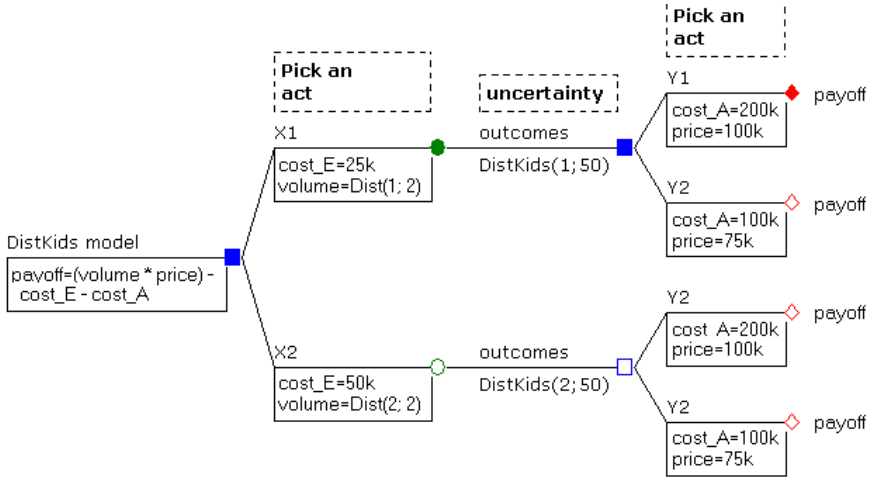
► **To dynamically distribute branches at a chance node:**

▷ Choose **Values >** Distributions…, and define the desired distribution, noting its integer index.



▷ In the probability for a chance node with a single branch, enter `DistKids(index; n)`. The first function argument is the distribution's index, and the second argument is the number of virtual "branches" to create internally during calculations.

▷ In a payoff or other expression, enter `Dist(index; 2)`. During any calculation that includes a branch generated dynamically by a DistKids() function referencing the same distribution, this usage of the Dist() function will return the parameter value sampled for that branch.



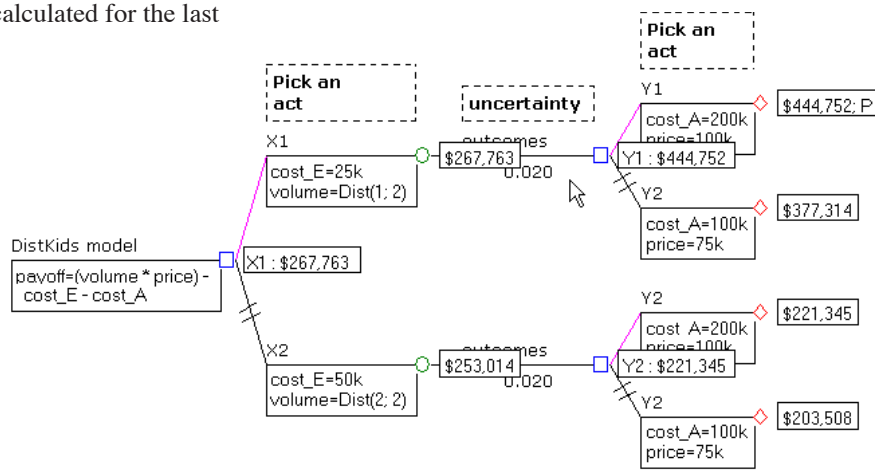
The behavior of the function during *any* calculation (not just Monte Carlo simulation) is to randomly sample *n* values from the selected distribution, creating *n* virtual branches (each with equal probability, 1/n, and each assigned a randomly sampled value).

Roll back displays the single visual chance node branch with a probability of 1/n, where n is the number of branches specified in DistKids(). The roll back box for the chance node using the DistKids() probability function will display the calculated mean for the n sampled branches, unless there is another chance node somewhere to the *left* that also uses the DistKids() function. Roll back boxes at nodes to the *right* of a chance node using the DistKids() function should be ignored, because they will display the value calculated for the last replicate of that node.

The value of the chance node is the average of the values of the *n* replicates of the tree structure to the right, with each replicate calculated using a sample value for the parameter.

TreeAge Pro incorporates the expected value of the DistKids() chance node into tree calculations to the left (for example, the initial decision node in the example), and also generates a special text report for each dynamic chance node.

The report includes rows showing each parameter sample value and the corresponding virtual branch's expected value; if the branch is a decision node, a column also reports which alternative is optimal for that branch/sample.



Distributing a fixed set of chance node branches

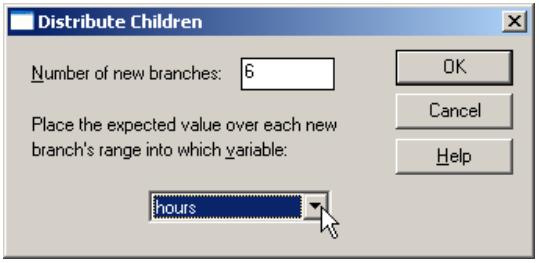
TreeAge Pro's Distribute Children command can be employed to assign real branches (not virtual ones, as in the case of the DistKids() function), probability estimates, and value estimates at a chance node, based on a selected distribution. The Distribute Children feature is useful when you would like to have a chance node that visually represents the impact that a Normal, Binomial, or other continuous or discrete distribution has on the model. This is a one-time process in which TreeAge Pro will generate the branches, probabilities, and parameter values associated with the specified distribution.

The Distribute Children feature is a static version of the dynamic, DistKids() feature described above. To use Distribute Children, you must have a chance node from which no branches emanate, and a variable to hold the values of the distribution in each branch. Unlike the DistKids() function, it is not necessary to create a distribution in TreeAge Pro's Distributions dialog prior to using the Distribute Children feature.

► **To distribute a fixed set of branches at a chance node:**

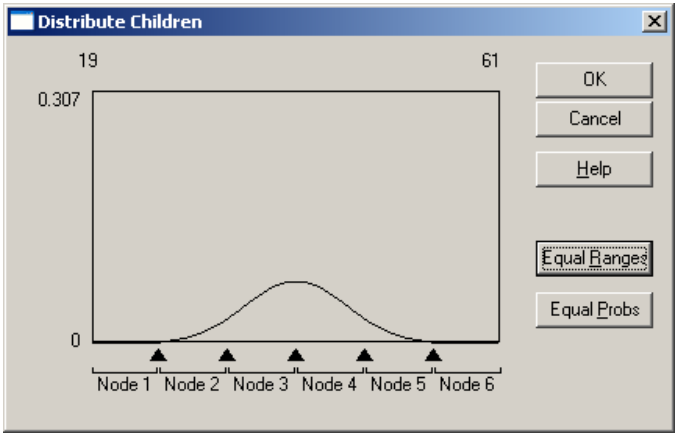
▷ Select a chance node with no branches, and choose **Options >** Distribute Children….

▷ In the first dialog, enter the number of branches you would like created, and select an existing variable to hold the values of the distribution. Click OK to continue.



▷ In the Distribution Picker dialog, select a distribution type and enter its parameters. Click OK to continue.

▷ In the final dialog, indicate how the branches should be distributed. See below for details.



You may also drag the handles which separate the nodes to create a custom allocation of the distribution.

▷ Press ENTER or click OK to create your distribution of branches.

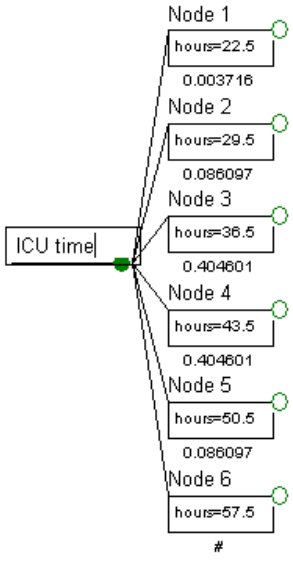
TreeAge Pro will create the branches for you. At each branch, a probability will be entered and the selected variable will be defined equal to a corresponding value from the distribution.

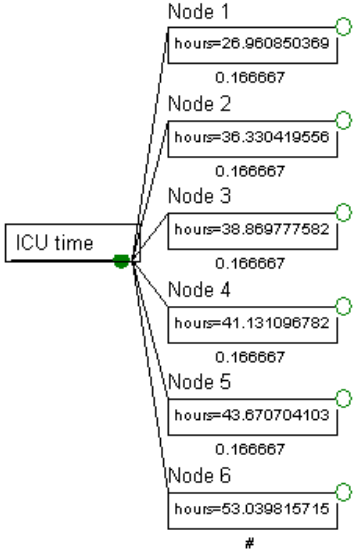
If you accept the default setting, to divide the distribution's range into equal segments, TreeAge Pro assigns each branch/node the segment's midpoint value and approximates the cumulative probability for the segment. If you click the Equal Probs button, TreeAge Pro instead starts at one end of the distribution and for each branch uses the target cumulative probability to determine an approximate segment (i.e., range of values). Again, a segment's midpoint is used as the value for the corresponding branch/node.

Compare the tree using equal probabilities, shown below right, to the tree using equal ranges, above right.

In some circumstances, the probabilities of the branches may add to more than 1.0 due to rounding error, for instance if there are numerous branches with small probabilities. If so, slight manual corrections to the calculated probabilities, or regenerating the subtree with a different number of branches or other options, will avoid an error message when you try to calculate the tree.

The custom, table-based distributions described in Chapter 17 cannot be used with the Distribute Children command.





Chapter 23

Bayes' Revision in Decision Trees

This chapter shows how, in the tree window, TreeAge Pro can assist you in performing the calculations that implement probability revision using Bayes' theorem.

Chapter 26 provides instructions on performing more complex Bayes' revision using influence diagrams.

In this chapter:

- An introduction to Bayes' revision.....310
- Performing Bayes' revision in the tree.....312

An introduction to Bayes' revision

If your model includes imperfect tests or forecasts followed by decisions, you may wish to utilize TreeAge Pro's Bayes' revision feature. TreeAge Pro can automatically perform probability revision using Bayes' theorem. The process occurs once, during the initial construction of the model; based on your answers to a few questions, TreeAge Pro will generate a set of variable definitions that calculate the revised probabilities. The probability expressions will be recalculated every time the model is evaluated, and results can change as your estimates of prior and likelihood probabilities (see below) change.

Bayes' revision is implemented in both the tree and influence diagram windows. Bayes' revision in the tree window is able to revise probabilities automatically based upon a single test. To revise probabilities associated with sequential tests, you should initially build your model as an influence diagram.

Probability revision using Bayes' theorem

Bayes' revision allows decision makers to calculate *decision* probabilities from *likelihood* probabilities. Likelihood probabilities, or forecast likelihoods, are answers to questions in the form, "If this test is performed on a part *known* to be faulty, what is the probability of a positive result, indicating a problem?" This type of probabilistic information is often available, but is not immediately useful in making decisions. What is needed are the decision probabilities, which address questions such as, "If a particular part tests positive, what is the probability that it really is faulty?"

The decision probabilities are so named because in the real world, they are the probabilities upon which decisions are based. These are also sometimes called posterior (or *a posteriori*) probabilities.

The basic formula for revising probabilities is:

$$P(H \mid E) = \frac{P(E \mid H) * P(H)}{P(E)}$$

where *H* is the hypothesis (e.g., faulty or not faulty) and *E* is the evidence (e.g., test result). The formula is applied once for each hypothesis-evidence combination — for example, P(not faulty | positive), or the probability not faulty given a positive test result. P(H) represents the prior (or *a priori*) probability of the condition. P(E) is a *marginal* probability, calculated as part of the revision.

A simple numeric illustration

The following example is designed to offer a sense of the potential usefulness of Bayes' revision. If you are already familiar with the type of applications that require Bayes' revision, you may want to skip this section.

Consider an automated test for a defect in a semiconductor. The defect is present in 1% of the items under scrutiny. It has been demonstrated that the available test will detect 98% of the faulty materials, meaning that 2% of those pieces with the defect will not be picked up by the test. Also, the test is known to incorrectly identify as faulty 3% of those pieces that are without defect.

You have under consideration installing a machine to perform this test in your facility. What is the likelihood that a part which tests positive actually has the defect? How certain can you be parts that tested negative don't have a defect? The information about the accuracy of the testing equipment provided above does not directly answer these crucial questions.

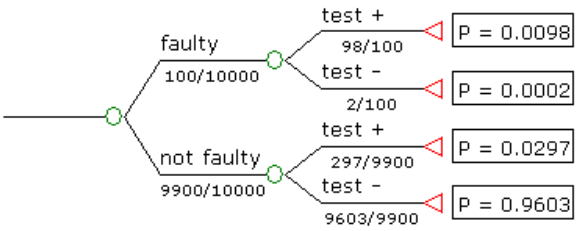
Let's say we have a batch of 10,000 items to be tested.

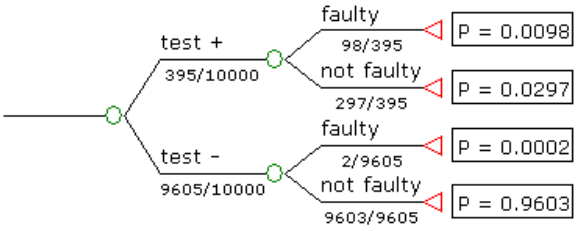
If the estimated prior probability of defect is 1%, we would expect 100 items in the batch to have the defect. Of these, about 98 should test positive. Of the 9,900 pieces without the defect, we said approximately 3% (297) would test positive.

Thus, a total of 395 (297 + 98) test subjects would test positive (this is one of the *marginal* test probability).

The Bayes' revision formula is intuitive when illustrated and worked out using a tree, as shown below.

The first revised decision probability is the ratio 98/395, or approximately 25%. This is the probability that a positive test actually indicates the presence of the defect. In this case, 75% of the positive tests are in error. The other decision probabilities are similarly calculated. With this information, a decision maker could compare the performance of the new test with existing methods or competing technologies.



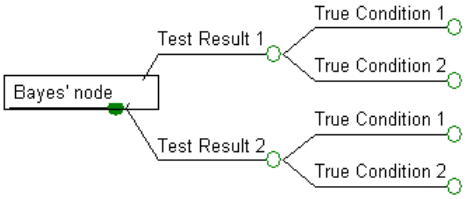


Performing Bayes' revision in the tree

Although the probability revision calculations can be done by hand, TreeAge Pro is able to manage this much more easily. Moreover, if the tree is set up using variables, it will be possible to carry out sensitivity analysis out on prior and likelihood probabilities.

To use TreeAge Pro's automatic probability revision, you should first obtain numeric values for the likelihood probabilities associated with the test and *the a priori* probabilities for the hypotheses. Then, your tree must be set up with a specific structure:

- The root of the Bayes subtree (the *Bayes node*) must be a chance node; it does not have to be the root node of your tree.
- The immediate descendants of the Bayes node must be chance nodes. They represent the possible outcomes of the test or forecast.
- The test subtree must be symmetrical, i.e. each test outcome (e.g., positive and negative) must have an identical subtree whose branches represent the possible underlying conditions (the hypothesis; e.g., condition present and condition absent).



Branches should be named descriptively. TreeAge Pro will walk you through the process of inputting the appropriate values or variables, and will need the branch descriptions in order to be able to phrase questions about the known and calculated probabilities.

Note that the structural limitations specified above apply only at the time of using TreeAge Pro's automatic Bayes' revision command. After TreeAge Pro calculates and inserts the decision probabilities, you may refine the structure of your model to include asymmetries or intermediate events, including decisions interposed between the evidence and hypothesis nodes. The structural requirement allows TreeAge Pro to decipher the natural structure of your problem during the probability revision calculations.

The limitations specified above do not necessarily apply when Bayes' revision is used in the influence diagram window; see Chapter 27.

Tip:

If you tell TreeAge Pro that you have sensitivity and specificity information for the test in question, you will need to indicate which evidence node represents a positive test result and which hypothesis node represents the presence of the condition for which you are testing.

Entering the probabilities

Once you have properly constructed the test subtree, you can choose its root node and perform Bayes' revision. TreeAge Pro will then ask a series of questions. If the test subtree represents a dichotomous test for a binary hypothesis (two possible results, two possible conditions), TreeAge Pro will ask if you have sensitivity and specificity information; see the Note below. Otherwise, for each hypothesis, TreeAge Pro will ask you to enter the *a priori* probability that the condition is present. It will then ask you to enter the test likelihoods associated with the hypothesis.

If you use variable names for the probabilities that you enter, you can easily perform sensitivity analysis on these parameters. It is also possible to enter numeric probabilities.

After you have completed the entry of probabilities, TreeAge Pro will create an additional set of variables representing the revised probabilities to insert into your tree. These variables are automatically set to hide their definitions (at the test node).

Let's get started with an example of Bayes' revision in the tree window.

▷ Open the file Oil Drilling Bayes, found in TreeAge Pro's Tutorial Examples subdirectory.

The probabilities have been left blank in this tree, shown below.

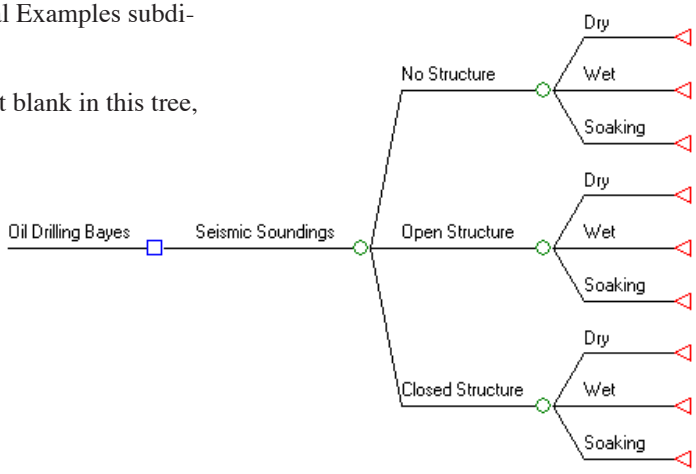
Simple prior probability estimates will be assumed for the outcomes of drilling an oil well in a target area based on prior success in the immediate region. We'll assume the prior results for 10 other wells, which will make up the prior probability distribution, are:

Dry:	5/10 (50%)
Wet:	3/10 (30%)
Soaking:	2/10 (20%)

Based on prior experience, we'll assume, a table is constructed showing for a well's actual state (dry, wet, or soaking) what the test probabilities are for seismic soundings (no, open, or closed structure) :

	No Structure		Open Structure		Closed Structure	
Given Dry	0.6	+	0.3	+	0.1	= 1.0
Given Wet	0.3	+	0.4	+	0.3	= 1.0
Given Soaking	0.1	+	0.4	+	0.5	= 1.0

If knowledge about the underlying geological structure of the well site can be obtained through seismic soundings, Bayes' theorem can be used to revise the probability distribution of the extractable oil deposits based on the test results.

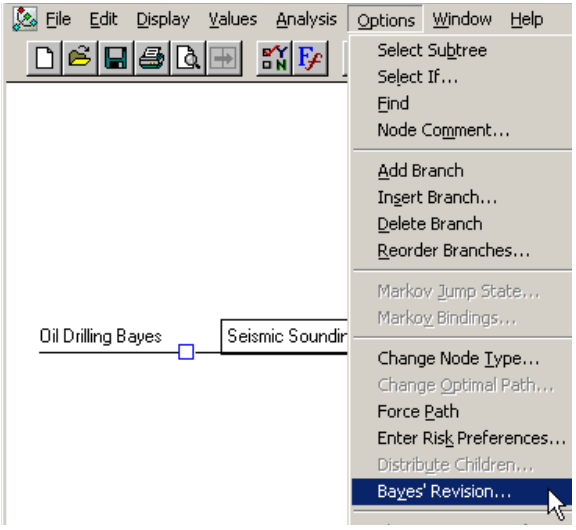


Take a moment to examine the structure of the Oil Drilling Bayes tree.

The extra decision node at the root will be explained later. The chance node closest to the root node has three branches representing the possible results of the seismic soundings. The results of the test are the first event in the tree. The ultimate condition of interest is the amount of oil that can be extracted, and this uncertainty is represented by the three subtrees with the branches Dry, Wet, and Soaking.

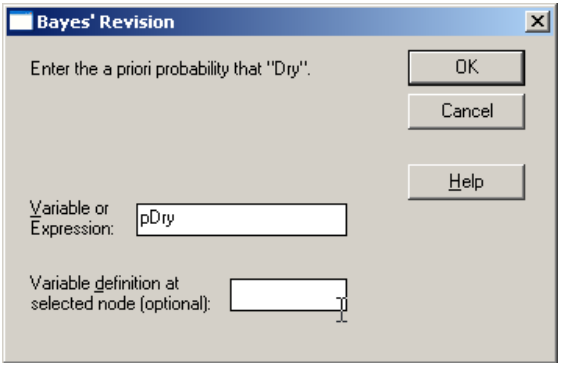
► **To perform Bayes' revision:**

▷ Select the Seismic Soundings node, and choose **Options >** Bayes Revision….



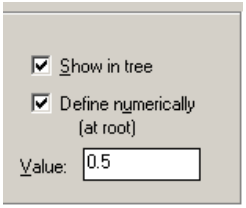
TreeAge Pro will present a series of dialogs for required parameters. In each dialog, TreeAge Pro requires a valid entry in the Variables or Expression field, usually the name of a new or existing variable. The bottom field is optional. It can be left blank; any value entered there will be used to define a variable specified in the top field.

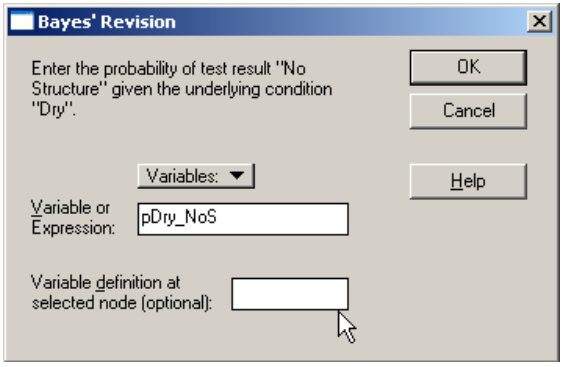
▷ For the prior probability of a dry well (your estimate prior to receiving the results of seismic sounding) enter the variable name `pDry` in the Variable or Expression field. Delete the 0 in the Variable definition… box. Click OK to continue.



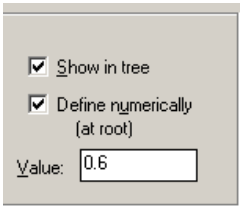
TreeAge Pro will prompt you to create the new variable pDry.

▷ Check the Define numerically option, and enter `0.5` in the value field. Click OK to create and define the new variable.





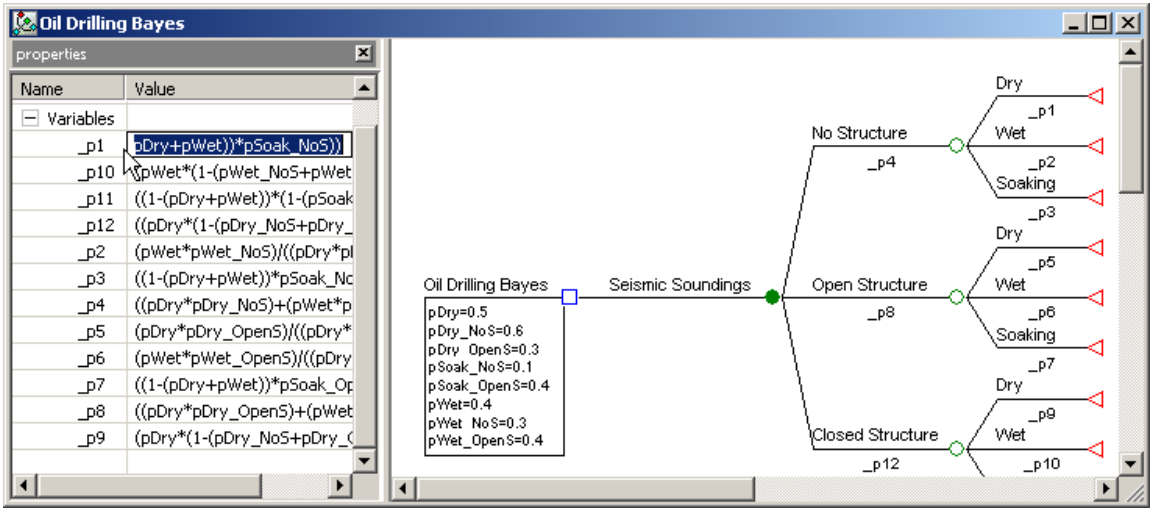
- For the likelihood probability that no structure will be detected in a dry field, enter `pDry_NoS` in the Variable or Expression field and click OK. Define the new variable equal to 0.6.



- For the probability that no structure will be detected in a dry field, enter `pDry_OpenS` in the Variable or Expression field and click OK. Define the new variable equal to 0.3.
- For the probability that closed structure will be detected in a dry field, accept TreeAge Pro's default expression 1-(pDry_NoS+pDry_OpenS).
- For the prior probability of a Wet well, use another new variable, `pWet`, and define it equal to 0.3. For the next two likelihood probabilities, use new variables `pWet_NoS` (= 0.3) and `pWet_OpenS` (= 0.4). For the third likelihood probability, accept the default expression again, 1-(pWet_NoS+pWet_OpenS).
- For the prior probability of Soaking accept the default expression 1-(pDry+pWet), which will calculate to 0.2. For the next two likelihood probabilities, use new variables `pSoak_NoS` (= 0.1) and `pSoak_OpenS` (= 0.4). For the third likelihood probability, accept the default expression again, 1-(pSoak_NoS+pSoak_OpenS).

Note that existing variables can also used in assigning likelihood probabilities, if you create them before performing Bayes' revision.

After the last required expression is entered, TreeAge Pro will define its own set of variables, at the selected test node, to hold the formulas for the decision probabilities.



The complex expressions used to define these variables (with names like "_p1") will be hidden from view in the tree. This is because TreeAge Pro automatically turns off the Show in tree property of these variables. See Chapter 13 for information on changing variable properties.

► **To view the definitions of the revised probabilities:**

▷ With the test chance node still selected, choose **Values >** Show Variable Definitions Window. Or, expand the Variables list in the properties grid, as in the above screenshot.

You may later perform sensitivity analysis on any of the variables you created to represent underlying quantities, both prior probabilities and likelihoods. You should *not* perform sensitivity analysis directly on the decision probabilities created by TreeAge Pro and used in the chance node branches. Analyzing the prior and likelihood probabilities, instead, will work correctly.

Making changes to the test subtree

Once you have completed the Bayes' revision process, it will probably be necessary to modify the structure of the tree in a number of ways. The example file Oil Drilling #3 illustrates a tree initially set up using Bayes revision, and then modified to include the various decisions. To get from the Oil Drilling Bayes tree to the completed Oil Drilling #3 tree, you must add the initial decision whether or not to undertake seismic tests, and the four decisions whether to drill.

In the Oil Drilling Bayes tree, you can select No Structure, Open Structure, and Closed Structure, one at a time, and choosing **Options >** Insert Branch. Choosing to insert a branch to the right of each will create new chance nodes.

Adding the initial decision, whether to undertake the seismic testing, requires close attention to detail. In this case, an extra decision node was included at the outset to simplify matters somewhat. The prior probability variables pDry and pWet should apply in both the Seismic Soundings and No Soundings subtrees; since they have been defined at the root node, this will be the case.

Valid prior and likelihood probability expressions

Two types of entries are valid in the Variable or Expression field. You may enter the name of a variable, as in the above examples, or you may enter an expression, such as "var1+var2." Expressions are scanned for new variables. You may *not* use the hash mark (#) in the Expression field.

Any variables already created in the tree may be selected from the pop-up menu and inserted into the expression.

If the expression entered is a simple variable name, the variable will be defined at the Bayes node if you enter a numeric value in the Variable definition field of the dialog. If you leave the definition field blank, as in the tutorial above, no definition of the variable will be added to the test node.

Node-specific definitions of test probabilities:

If, as in the example, the test node is an internal node, entering both a new variable name and a numeric value in the Bayes' revision dialog, and then giving the new variable a numeric value in the Properties dialog, may create definitions in two places: at the selected test node and at the root node of the tree.

If test probability variables are defined at both the root node and the test node, the root node (default) definition will be ignored in the Bayes subtree, in favor of the closer, test node definition. The danger is that later you may change the value at the root node definition, but not at the Bayes node definition, resulting in unexpected results.

Chapter 24

Utility Functions and Risk Preference

TreeAge Pro allows you to set up a risk preference utility function in a tree, which can be used to account for a decision maker's aversion to risk.

In this chapter:

- Risk preference: an illustration 320
- Risk preference curves 322
- Creating a risk preference function 323

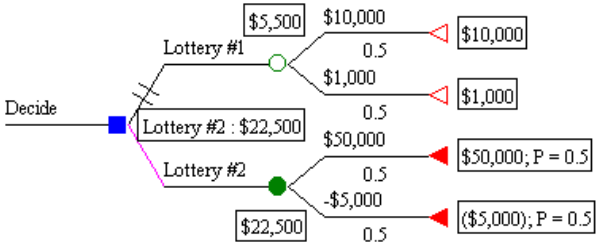
Risk preference: an illustration

Most decision makers are risk-averse to some degree. They are willing to pay a premium, small or large, to avoid risk. The decision maker's risk preference can be incorporated into a decision tree.

Assume that a rich uncle offers you an opportunity to win some money. He proposes to flip a coin giving you the opportunity to receive either $10,000 or $1,000, depending on whether you correctly predict the outcome. If you call the flip correctly, you will receive $10,000, and if you are wrong you will receive $1,000.

To make this game more interesting, assume that your uncle complicates matters by offering an alternative opportunity. The alternative is also a coin flip. Under this one, you will receive $50,000 if you are correct, but you will have to pay him $5,000 in the event you lose on the coin flip. There will be only a single coin flip; it is up to you to choose between the two. As you will see, it may not be wise to base your decision solely on traditional expected value calculations.

The tree below models your uncle's offer.



As the tree illustrates, there are two lotteries. Both provide the same (50 - 50) odds of winning, but they have different outcomes. You must choose one of them. On the basis of expected value, you should choose lottery #2. Its expected value ($22,500) is more than four times that of lottery # 1 ($5,500).

However, what about the risk posed in lottery #2 that you could actually end up losing $5,000? At least in lottery #1 there is no risk of being out-of-pocket – you are guaranteed to win something. How one responds to the downside risk posed by lottery #2 involves a subjective analysis of the decision maker's *aversion to risk*.

Certainty equivalents and risk aversion

Consider lottery #1 described above. The expected value is $5,500. Would you sell the opportunity to play this lottery for $4,000? If you were offered $3,000 by a third party who wanted to buy into the lottery, would you sell?

The minimum value for which you would sell the lottery is your *certainty equivalent* for this lottery. The certainty equivalent of a lottery can be perceived as the expected value of that lottery, adjusted for risk preference (the *risk-adjusted* expected value).

A certainty equivalent is similar to an expected value, in that it is a single numeric quantity which represents the value of an uncertain event. The certainty equivalent is a subjective measure. It is the answer to a question of the form, "What is the minimum (or maximum) value for which I would trade this uncertainty?"

Now consider a situation which is undesirable from the start. Lottery #3 is a coin flip in which you will either owe your uncle $2,000 or you will owe him $12,000. In this situation, we are interested in finding the *maximum* amount that you are willing to pay to a third party to assume your obligation under the lottery. Would you pay $4,000? Or $5,000? Your answer to this question is your certainty equivalent for that lottery.

The certainty equivalent for a lottery is usually in the same numeric range as the expected value. The gap between the certainty equivalent and the expected value is a measure of *risk aversion*.

Most decision makers are risk-averse to some degree. They are willing to pay a premium, small or large, to avoid risk. Their certainty equivalent for any lottery will be lower than the lottery's expected value. In contrast, a risk-seeking decision maker is one whose certainty equivalent for a lottery is higher than the lottery's expected value. The risk taker is willing to pay a premium in order to participate in the lottery.

Risk preference curves

A straight-line risk-preference curve represents a decision maker who is *risk-neutral*. This type of decision maker bases decisions on expected values rather than certainty equivalents.

A risk-averse decision maker will have a curve with a decreasing slope, meaning that certainty equivalent is less than expected value. The curve will typically be steeper in the low value range, where aversion to risk is weak, and will grow progressively flatter as the values get larger (both positive and negative), where aversion to risk becomes stronger. The more risk-averse you are, the more your curve will deviate from the 45° straight line representing risk neutrality.

If you encode a curve that includes some unexpected bumps when graphed, this means that some of your responses were inconsistent. You should repeat the process. Don't be discouraged; developing a meaningful non-constant risk utility curve takes hard thinking and careful consideration.

For further reading on risk preference, refer to the general texts on decision analysis listed at the end of Chapter 1.

Note:

Risk preference functions can be used only if the calculation method is set to Simple.

Creating a risk preference function

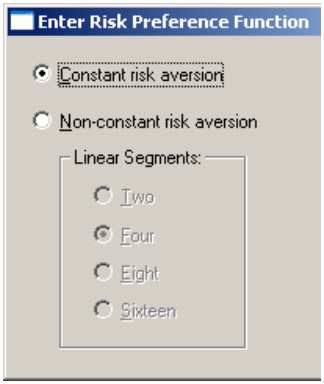
TreeAge Pro is able to record your risk function as a mathematical curve, and apply this curve to the expected value of an uncertainty. Recommendations are then made based on your derived certainty equivalents, rather than on expected values.

There are two types of curves, or risk functions, which TreeAge Pro can use. The *constant risk-aversion* function is calculated using the formula:

$$U(x) = 1 - e^{-x/R}$$

where U is an arbitrary utility scale, and R is a risk preference coefficient, described below. The utility scale is used only for internal calculations; the formula's inverse is later applied to find certainty equivalents.

The *non-constant risk function* is tailored to fit your specific model, and so is superior to the constant risk aversion function in many respects, except that it takes a little longer to set up initially. TreeAge Pro will ask you a series of questions about your certainty equivalents for the model you are working on. It will then create a curve made up of line segments approximating your true risk function.



Constant risk aversion

If constant risk aversion is selected, you will be asked to supply a single value. Specifically, you will be shown a generic lottery in which you have a 0.5 probability of winning *X* and a 0.5 probability of losing one-half *X*, and asked to specify the largest value of *X* for which you would be willing to take part in the lottery. This value is used as the risk preference coefficient in the above formula.

The lottery might represent an investment in a biotech company which is about to get a judicial ruling on the validity of an important patent. If the ruling is favorable (0.5 probability), the investment will double in value; if unfavorable (0.5 probability), the investment will fall in value by 50%.

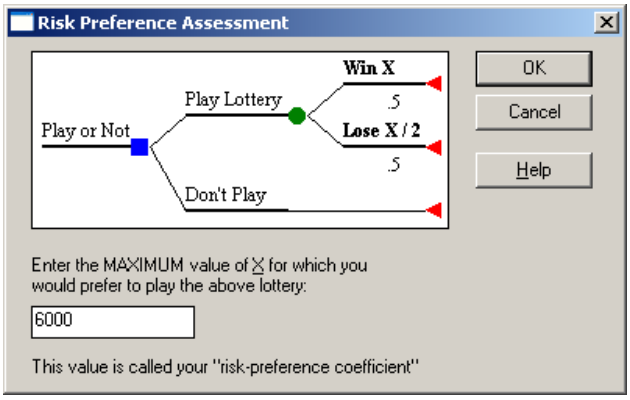
What is the most you would invest under these circumstances? This amount is referred to as your risk preference coefficient.

► **To assign a constant risk-aversion preference function:**

▷ Choose **Options >** Enter Risk Preferences… (or Edit > Preferences…, and select the Risk Prefs category).

▷ Ensure that Constant risk aversion is selected, and choose OK.

▷ Answer the single lottery question, shown below. This is the risk preference coefficient.



Non-constant risk aversion

Because the non-constant risk function is assessed on the basis of the range of payoffs in your model, it is often not possible to set up this risk function until after your model is complete.

After entering the non-constant risk function, it can be graphed from the **Analysis** menu; the graph's text report will show the values used in the function.

► **To assign a non-constant risk-aversion preference function:**

▷ Select the root node and choose **Analysis** > Payoff Range, to ensure that your model calculates correctly (to create the risk preference function, TreeAge Pro must determine the range of potential payoff values).

▷ Choose **Options** > Enter Risk Preferences….

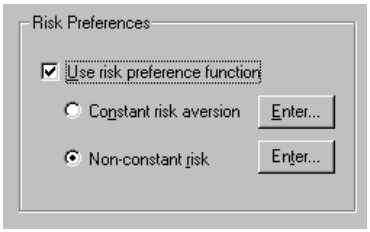
▷ Choose Non-constant risk aversion. Enter the number of linear segments used to approximate your preference function. More segments will result in a more accurate picture of how your risk preferences vary over a range of potential outcomes, but you must take the time to answer more questions about your certainty equivalents.

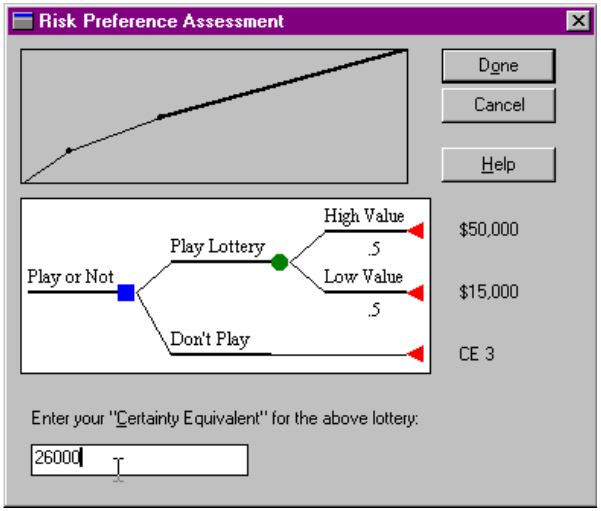
▷ TreeAge Pro will ask you for certainty equivalents for a series of lotteries based on your tree. For each lottery, you must assign the minimum (or maximum) certain value for which you would trade the uncertainty displayed.

▷ Click the **More** button after entering each certainty equivalent. For the last certainty equivalent, click Done.

▷ As you supply the requested information, TreeAge Pro will generate your risk preference curve, shown in the top part of the window.

If you plan to change any of the payoffs in your model after entering a non-constant risk preference function (including during a sensitivity analysis), note that the risk function may no longer be valid if payoff values fall outside the range of payoffs considered in the risk preference function. TreeAge Pro will warn of an attempt to apply an invalid non-constant risk preference function.





Risk preference notes:

- Risk preferences can also be set up via the Preferences dialog box. From there you may turn the risk preference function on or off, and enter the risk preference functions.
- The active risk preference function can be graphed by choosing **Analysis >** Graph Risk Preference Function.
- If you are using the constant risk-aversion preference function, you can perform a one-way sensitivity analysis on the risk preference coefficient. At the bottom of the list of variables specified in the sensitivity analysis dialog box will appear Risk Preference Coeff. If you select this "variable," your analysis will graph the effect of varying the coefficient's value.
- When a risk preference function has been turned on, an item in the status bar will read RISK. In addition, the boxes that appear to the right of each node following Roll Back will be drawn with rounded corners.
- A problem of the constant risk aversion function is that the accuracy of the exponential formula on which it relies becomes unreliable (or mathematically undefined) if any one of the model's outcomes — with either a positive or negative payoff — are much larger than the risk preference coefficient. The non-constant risk-aversion function avoids this problem.
- To extend the range of payoffs used in building the risk preference function, add a zero probability event somewhere in the tree with the highest (or lowest) possible value to represent in the function.

Working with Influence Diagrams

The two chapters in Part V detail TreeAge Pro's influence diagram interface. These chapters cover the basics of influence diagrams as well as advanced topics such as asymmetry, Bayes' revision, and EVPI.

See Chapter 29 in Part VI for information on using the TreeAge Pro Excel module to convert a spreadsheet into a fully-linked influence diagram.

In this part:

- Chapter 25, "Building Influence Diagrams – Tips and Tools"
- Chapter 26, "Advanced/Intermediate Influence Diagram Topics"

Chapter 25

Building Influence Diagrams – Tips and Tools

This chapter builds on the basic information provided in Chapter 3 on creating influence diagrams in TreeAge Pro. The topics covered in this chapter – including creating asymmetry, converting influence diagrams to trees, and using value nodes – will apply to most influence diagrams you create.

See the Excel module documentation for information on importing fully-linked models from spreadsheets.

In this chapter:

- When to use influence diagrams 330
- TreeAge Pro's node conversion algorithm 331
- Asymmetry 333
- Deterministic nodes, value nodes, and node variables..... 334
- Using the assessment window 337
- Miscellaneous..... 340

When to use influence diagrams

TreeAge Pro's influence diagrams has both benefits and drawbacks compared to trees, which may affect your decision about whether to begin a particular model as an influence diagram. First, the benefits:

Bayes' revision

TreeAge Pro's implementation of influence diagrams includes some features that simply are not available in the tree window. Influence diagrams can, for example, automatically calculate posterior probabilities using Bayes' revision with multiple sequential tests (the tree window's implementation of Bayes' revision can handle only a single test). Moreover, TreeAge Pro's implementation of Bayes' revision is handled more intuitively in the influence diagram window. For example, in the tree window, you must perform Bayes' revision prior to modeling any intervening decisions. Since this restriction does not apply to influence diagrams, several steps in the model-building process can be avoided.

EVPI (expected value of perfect information)

Calculation of EVPI can be handled more elegantly in the influence diagram window. There are situations that the tree implementation of EVPI cannot handle, especially in larger models, but that are straightforward and simple to handle in the influence diagram window. Detailed instructions on implementing Bayes' revision and EVPI in the influence diagram window follow in Chapter 26.

Model size and other considerations

Many model builders feel that the ability to build the model as an influence diagram which can be converted automatically into a fully-functioning, asymmetrical tree offers the best of both worlds. With its compact size, it is often more practical to print and present an influence diagram than the associated tree. Moreover, building an influence diagram forces you to consider issues of influence that may be overlooked when building a tree.

While learning to build influence diagrams, the greatest frustration flows from uncertainty about whether their influence diagram correctly models their problem. In TreeAge Pro, the remedy is to convert the influence diagram into a tree, and check the tree's structure. If the tree is not what you expected, return to the influence diagram, correct it, and try again.

Limitations of an influence diagram

Certain features – for example multi-attribute (such as cost-effectiveness) models and Markov processes – are not available in influence diagrams. This doesn't mean that a model begun as an influence diagram can never contain these features, but simply that they will have to be added after the influence diagram has been converted into a tree.

As the model builder, you will have to decide, on a case-by-case basis, whether it pays to begin a model as an influence diagram when it cannot be converted into the completely functional tree you require.

TreeAge Pro's node conversion algorithm

In TreeAge Pro, it is not possible to perform calculations directly on an influence diagram. To calculate the model, the influence diagram must be converted into a tree. Thus, to correctly structure an influence diagram, you should know the rules by which TreeAge Pro will convert that influence diagram into a tree. These rules conform to the standard method of influence diagram construction and conversion.

> ***Tip:***
> How the standard conversion algorithm makes it possible to automate Bayes' revision and EVPI is covered in Chapter 26.

- **Decision nodes are ordered.** TreeAge Pro determines in what order decisions occur by looking at the arcs between them. You should draw arcs from a decision node to every other decision node which occurs later in time. This convention not only resolves any ambiguity regarding decision ordering, but also conforms to the standard use of *no-forgetting* arcs (see tip below). Any remaining ambiguity in ordering the decisions is resolved by graphical node position, as described below.
- **Chance nodes are grouped.** For each chance node, TreeAge Pro determines which decision nodes it precedes and which it follows. This determination is made solely on the basis of direct arc-connections. If there is an arc from the chance node to a decision node, the chance node must precede the decision, and all subsequent decisions. If there is no arc, or if the arc points from the decision node to the chance node, the decision node will be converted first. You should draw arcs from a chance node to every decision it precedes. If the outcome of an uncertain event is known when the decision is made, the uncertainty precedes the decision. This also conforms to the *no-forgetting* principle of arcs. At this point, chance nodes will be grouped in positions between decisions, although no ordering has taken place within each group. Some chance nodes may, of course, precede or follow all decisions.
- **Chance groups are individually sorted.** For each group of chance nodes, the order is determined by considering arc flow, with a node at the base of an arc converted before the node at the tip. Ambiguities are resolved by considering graphical position, as described below. Each group is ordered independently; for this purpose, arcs from nodes in one group to nodes in another group are ignored.

No-forgetting arcs

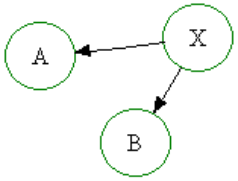
The no-forgetting principle of arcs simply states that if a node precedes a decision, it must also precede all subsequent decisions. Thus, the information is remembered at all subsequent decision points. In TreeAge Pro, no-forgetting arcs are not required, so long as you draw an arc from each decision or chance node to its immediate successor decision node. However, in order to avoid confusion when sharing your model with others, you may want to include arcs to all subsequent decisions.

To create a timing-only arc, double-click on an existing arc, and uncheck all of the Probs and Values check boxes in the Arc Info window, and ensure that all the structural influences are set to "Symm" (symmetric).

Using graphical position to resolve time order

Some timing issues may not be resolved by considering arcs alone. In these cases, TreeAge Pro will convert nodes on the left of the influence diagram before nodes on the right. (There is a preference item which will force this left-to-right ordering to be performed top-to-bottom.)

For instance, consider the following influence diagram fragment.



The arcs indicate timing, so X will be converted before both A and B. Since the arcs provide no way to tell between A and B which should appear first in the tree, their respective positions in the influence diagram are used to make this determination. Since A is to the left of B, the nodes will be converted X, A, B.

The left-to-right (or top-to-bottom) ordering of nodes uses the *center* of each node as the point by which sorting occurs.

Asymmetry

This topic was covered initially in Chapter 3. If you have not worked through the tutorial in that chapter, it may be helpful to do so now.

Here is a description of the various structural influence types and how you might use them:



- **Symm -** Short for symmetric, Symm indicates that, for the influenced event, the tree should be as bushy as possible, with all branches drawn.
- **Force -** Use this influence type to indicate that when one *conditioning* event occurs (or one alternative is chosen), the result (or choice) associated with the *conditioned* node is known or determined. You will need to pick which outcome is forced via a pop-up menu in the Additional Info field.
- **Elim -** Similar to Force, this command eliminates one possible outcome or alternative from the conditioned node. You will need to pick which outcome is eliminated via a pop-up menu in the Additional Info field.
- **Skip -** This is the most common type of structural influence. It indicates that when one outcome occurs (or one alternative is chosen), all branches associated with the *conditioned* node should be omitted; events that follow the skipped node will still be included in the model.
- **Skip All -** Use this to indicate that *all* subsequent events are to be eliminated. This is a shortcut for creating many Skip arcs to other nodes in the influence diagram. Whenever the particular outcome or alternative is reached, it will become a terminal node in the tree.

If an arc has no probabilistic or value influence, and is used only to indicate asymmetry, it is drawn in dotted gray. You may choose not to print these nonstandard arcs; see the section describing preferences below.

An example

Consider, for example, a patient with an early stage of prostate cancer; you must decide whether to (i) perform a prostatectomy, (ii) treat with radiation, or (iii) do nothing. If you choose to do nothing immediately, the outlook for the patient is uncertain (with possible outcomes "Metastasized Cancer," "Local Cancer," and "No Detectable Problem"). However, if you choose to remove the prostate, then the outcome will always be "No Detectable Problem." In some cases, this asymmetry can be handled with the "Skip" command, but in many situations you will want to "Force" the event with a single branch (e.g., "No Detectable Problem") for clarity.

The "Elim" option might be used to indicate that if radiation therapy is the selected treatment, one of the possible outcomes, "Metastatic Cancer," may be eliminated. The situation is still uncertain, but one possibility has been eliminated.

Deterministic nodes, value nodes, and node variables

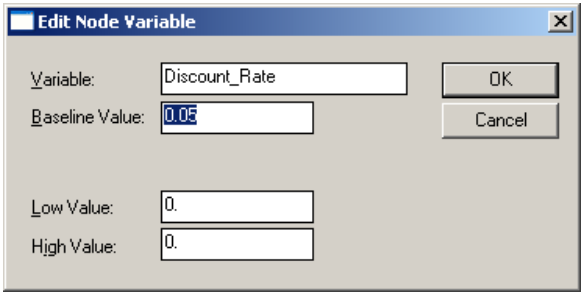
This section describes how to use value nodes, and deterministic nodes to create complex payoff formulas for your models. It also covers the use of node variables at chance and decision nodes.

Deterministic nodes

A deterministic node is useful for including a parameter which has a single, fixed value in your model, even if your estimate is uncertain. A deterministic node is drawn with a purple double-line.



You have the option of assigning not only a numeric value to a deterministic node, but also a range of possible values for future sensitivity analysis. (TreeAge Pro also allows *non-numeric* expressions to be entered for the Baseline Value.)



In other words, deterministic nodes in an influence diagram are used in much the same way that variables are used in a tree. Rather than entering a fixed numeric value, use a deterministic node to allow for more complete analysis. The variable associated with the deterministic node may then be used in the probability or value associated with whatever node(s) it influences. See the section Using the assessment window for more details.

Variables created from deterministic nodes will be defined at the root node of the converted tree.

Value nodes

Every influence diagram must contain a value node (even if payoffs are left empty).

There are two different uses for value nodes. The primary purpose of value nodes is to act as a placeholder for the model's final outcome measure (e.g., "Profit"). The final value node has arcs leading in from the nodes which influence the payoff, but has no arcs leading out. Payoffs may be enumerated on a scenario-by-scenario basis, or described by a formula.

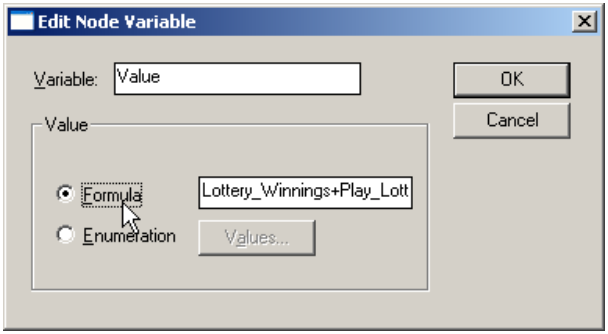
The secondary use of value nodes is to create an intermediate formula or payoff calculation (e.g., "Costs" or "Revenue"). Arcs would point to this intermediate value node from influencing events. This value node would then have an arc pointing to the final value node. Intermediate value nodes may be described only by formula, not enumeration.

If you assign a formula to a value node variable, the formula will be defined at the root node of the converted tree.

► **To assign a formula to a value node:**

▷ Select the value node, and choose **Diagram >** Variable… (or use the right-click menu).

▷ Select the Formula radio button.



▷ Enter the formula in the editor. Use the Insert pop-up menu to select variables from other, influencing nodes.

► **To enumerate payoff values for a final value node:**

▷ Select the value node, and choose **Diagram >** Variable (or choose Variable from the right-click pop-up menu).

▷ Select the Enumeration radio button.

▷ Click the **Values…** button to bring up the mini-tree value assessment dialog.

Enumeration is the default value assignment for final value nodes.

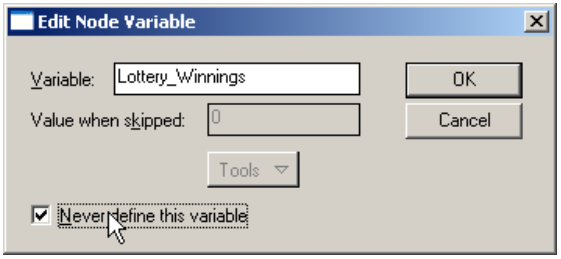
Node variables

Each node in an influence diagram can represent a parameter in your model's calculations. TreeAge Pro will automatically generate a conforming variable name from the node name when you first create it. You will be able to modify the variable name at any time. See Chapter 13 on variable naming rules in TreeAge Pro.

A variable associated with an influence diagram node may be used only to calculate a payoff formula. Node variables cannot be used in probability calculations; see the section Using the assessment window. Not every parameter in your model will be a part of a payoff formula. Accordingly, it is possible to suppress a selected node's variable.

► **To indicate that a node's variable will not be used:**

▷ Select the node, and choose Variable from the right-click pop-up menu (or from the Diagram menu in the main menu bar).



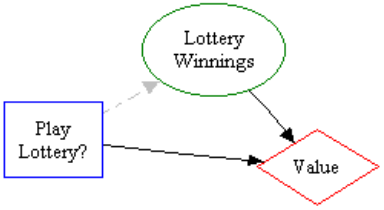
▷ Check the box labeled Never define this variable.

Setting this flag will prevent the node's variable from being defined in the tree. As a result, the node's variable will not be available for use in the payoff formula, and the node will not be available for value-conditioning by another node (no arc pointing to the node may indicate a "values" influence type).

The use of node variables in formulas is discussed below in the context of value nodes.

Node variables and asymmetry

The model shown at left illustrates asymmetry. The value of the model (i.e., the payoff formula) is the difference between the value of the lottery and the cost of playing. If, however, you decide not to play the lottery, and the uncertainty is skipped, how will TreeAge Pro determine the value of the Lottery

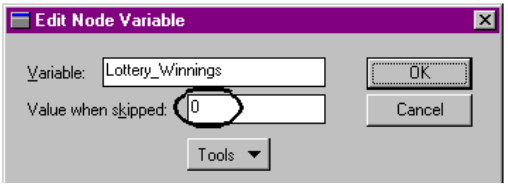


variable for purposes of calculating the payoff?

If a node whose variable is used as part of the payoff formula is skipped by asymmetry, TreeAge Pro offers you the chance to give it a default value. For instance, if you decide not to play the lottery, you would give the lottery a value of 0 whenever asymmetry indicates that the node should be skipped.

► **To assign a default value to a variable:**

▷ Select the node, and choose Variable from the right-click pop-up menu (or from the Diagram menu in the main menu bar).



▷ Enter a default value in the box labeled Value when skipped.

Using the mini-tree assessment window

The assessment mini-tree window, first described in Chapter 3, is used to assign both probabilities and variable (or payoff) values. Its usage is virtually identical in both cases. Each node which requires your attention will be displayed with a red diamond. Note that while these nodes look like terminal nodes in this *mini-tree*, they are not necessarily final outcomes.

The first red diamond node, at the top of the mini-tree, will be selected initially. A selected node will have its diamond filled in and its name drawn in bold. Only red-diamond nodes may be selected. When a node in the mini-tree is selected, the text editor in the right part of the dialog becomes active. The information you enter in the text editor will be used for the selected node.

To select another node in the tree, you may use any of the following methods:

- click directly on the node in the mini-tree;
- use the Prev/Next buttons to select the next node in the indicated direction; or
- hold down the CONTROL key and press either UP ARROW or DOWN ARROW.

To view more or less of the mini-tree, use the Zoom pop-up menu (displayed with a magnifying glass icon) or resize the window itself.

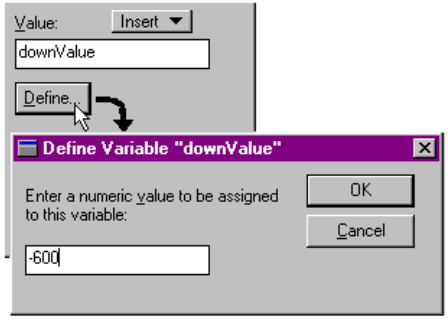
Entering a new variable in the mini-tree editor

The text editor accepts any valid number or expression. You may enter numeric values directly, such as 0.4, or you may enter expressions, such as *1-pLow*, *Cost/(1+rate)^time*, or #. You may also define a new variable.

Unlike deterministic nodes, in which the variable is always defined at the root of the converted tree, variables created in the assessment window are defined at the selected node or its parent. This may result in multiple (identical) definitions in the tree.

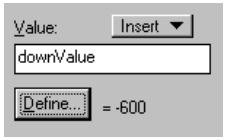
► **To define a new variable in the assessment window:**

▷ Type the name of the variable in the text box and click the **Define…** button.



▷ Enter a numeric value in the Define Variable dialog box, and press ENTER.

The numeric value is displayed next to the Define button, as well as in the mini-tree.



▷ To change the definition of the variable, click Define… again.

▷ To eliminate the definition, simply type over the variable's name in the editor. TreeAge Pro will use the new expression and eliminate the old variable.

Using existing variables in the mini-tree editor

The Insert pop-up menu enables you to use variables from other, influencing nodes. You are urged to use variables in the editor box rather than assigning numeric values directly. These variables may be defined either in the editor box (as just described) or at a separate deterministic node. Entering variable names directly in the editor (regardless of which method you use to define the variables) will encourage you to assign different variable names to different outcomes, such as "CostLow" and "CostHigh," making it very simple to perform sensitivity analysis on your parameters. (See Chapter 13 for more on this subject.)

If, instead, you enter numeric values directly, you run the risk of having multiple numeric definitions of the same node variable. In other words, the node's main variable will be defined at each branch with each of the different numeric values in turn; this will cause analysis problems later.

The mini-tree probability wheel

The probability wheel is available when editing probabilities for a chance node with at least two but not more than seven outcomes. The probability wheel and its use are described, in the context of decision trees, in Chapter 12.

► **To use the probability wheel in the mini-tree:**

▷ Select one of the branches of a chance node. The wheel operates on the *set* of branches, so it does not matter which of the branches you select.

▷ From the Tools pop-up menu, choose Probability Wheel.

Insert/create distributions in the mini-tree

The Distribution option in the Tools pop-up menu can be used either to assign values (e.g., payoffs) or probabilities. New distributions can be added to the list in the influence diagram, and then selected for use in either a single payoff, or a set of branch probabilities. The Distributions list can also be viewed and modified within the influence diagram window, via the **Values >** Distribution command.

The creation and management of distributions are described, in the context of decision trees, in Chapter 16.

► **To use a distribution in the mini-tree:**

▷ Select one of the terminal nodes, or branches of a chance node.

▷ From the Tools pop-up menu, choose Distribution…. Use the Distribution dialog to create a distribution, if necessary, and click Use to insert a reference to the distribution in the node editor.

Distributions that you create in the influence diagram will be included in a converted tree.

Dynamic linking in the mini-tree

It is possible to link values or probabilities to spreadsheet cells while in the influence diagram.

► **To link to a spreadsheet cell value:**

▷ Switch to the spreadsheet application, select a named cell to link with, and choose **Edit >** Copy.

▷ Switch to TreeAge Pro. Select a node in the influence diagram and choose **Diagram >** Probabilities or **Diagram >** Values.

▷ In the mini-tree window, select the node which will use the link. From the Tools pop-up menu, select Paste Link.

You can then edit the list of links using the **Edit >** Dynamic Links dialog. When you convert the influence diagram into a tree, the links will be included.

See Chapter 30 for instructions on using the Excel module to convert a spreadsheet into a fully-linked influence diagram. See Chapter 20 for more information on creating and using links.

Miscellaneous

Note boxes

Annotation can be added to the face of the influence diagram using note boxes. Simply right-click on an empty space in the influence diagram, and choose Create Note. Click and drag the crosshairs pointer to make a sufficiently large text box to contain your text. You can resize the box later, if needed.

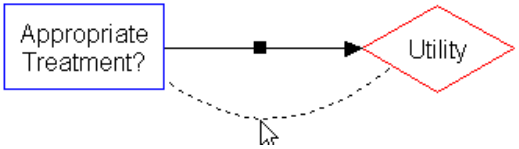
The creation and use of note boxes are described in Chapter 10, in the context of decision trees.

Note boxes are *not* included in the converted tree.

Node description

Nodes in the influence diagram may also be annotated, with hidden notes. This type of annotation is *not* carried over to the converted tree.

Select the node you wish to annotate, and choose **Diagram >** Description. In the Node Description dialog, both the node name (as it displays in the window) and the hidden annotation may be changed.

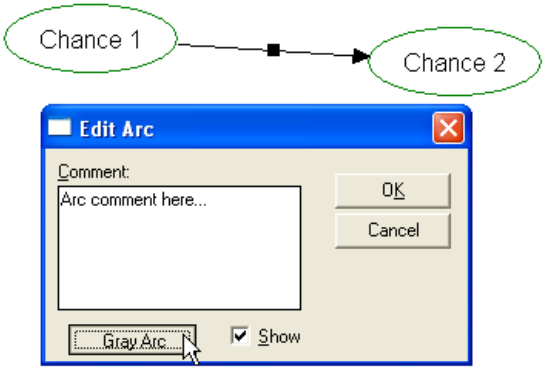


Arc operations

You may curve a selected arc by clicking on it and dragging its square handle.

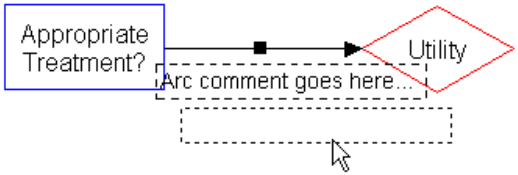
To straighten a curved arc, select it and choose **Diagram >** Straighten Arc (or right-click on the arc and choose the same command from the quick menu).

To flip a selected arc's direction, choose **Diagram >** Flip Arc. Any asymmetry specified in the arc will be lost, as the direction of influence has been reversed. This operation is primarily graphical; it does not perform intelligent reassignment of probability or value information. However, careful arc-flipping can help you calculate EVPI in the converted trees. See Chapter 5 for more on EVPI.



The arcs between any two nodes can be grayed, to indicate special conditions.

Arcs may be annotated. Double-click the arc (or right-click on it and choose Arc Info…) and enter a comment in the dialog. If the comment displays in an undesirable location, it can be moved by clicking anywhere inside the comment and dragging.

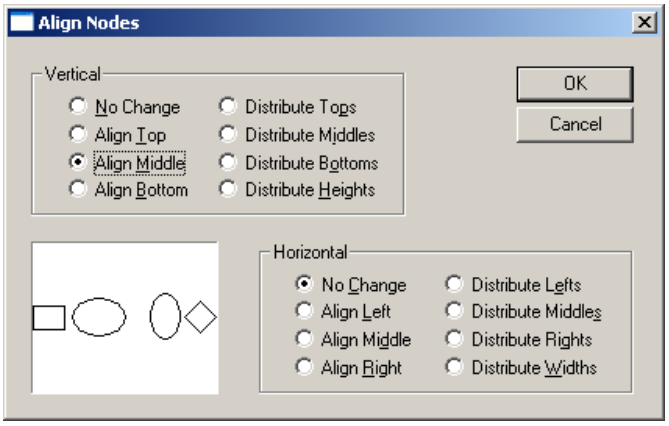


Arcs may *not* be cut and pasted like other objects, because the information stored in an arc is specific to the nodes that the arc connects. If you copy two nodes linked by an arc, then the arc will be copied (and pasted) with the nodes.

To eliminate a selected arc, choose **Edit >** Clear Arc as described in Chapter 3.

Aligning nodes

When your influence diagram is ready for presentation, you may wish to align nodes so that arcs appear perfectly straight. To accomplish this, select those nodes you wish to align, and choose **Display** > Align.



If you vertically align the centers of your nodes, then arcs between them will be drawn perfectly horizontally. If you horizontally align the centers of your nodes, then arcs between them will be drawn perfectly vertically.

Remember that, during conversion, ambiguous time-ordering is resolved on the basis of node position; in particular, the position of the *centers* of the nodes is used to sort them. If ordering is performed left-to-right, and two nodes have been horizontally aligned at their middles, so their horizontal position is identical, then the resolution of time-ordering is unpredictable. In situations such as this, you should use a visible, even exaggerated, horizontal displacement to indicate time-ordering.

Chapter 26

Intermediate/Advanced Influence Diagram Topics

This chapter covers a number of influence diagram features that may be important in some models.

In this chapter:

- Bayes' revision in influence diagrams 344
- EVPI (expected value of perfect information) 348
- Clones 349
- Sub-models 350

Bayes' revision in influence diagrams

Bayes' revision constitutes an exception to the rule that all of TreeAge Pro's calculations are performed in the tree window. Since Bayes' revision involves not only calculations but also elements of model design, TreeAge Pro performs all calculations associated with Bayes' revision during the process of converting your influence diagram into a tree.

Accordingly, you must enter the likelihood probabilities (such as the probability of a positive test result on a patient known to have a certain disease) within the influence diagram window. Based on this information, TreeAge Pro will calculate the decision probabilities (such as the probability that a patient has the disease if she has tested positive) at the time that the influence diagram is converted into a tree.

If you set up your tests properly in the influence diagram window, you will not need to use the Bayes' revision tool in the tree window, as all of the revised probabilities' formulas will have been created for you.

A copy of the tree built in the following tutorial can also be found in TreeAge Pro's Tutorial Examples subdirectory, with the name "Bayes ID". You may want to check your results against that file after working through the tutorial.

The problem being modeled involves deciding whether or not to replace the product of a manufacturing process. If the machine is faulty, it should be replaced. If it is not faulty, the cost of replacement would not be warranted. A reasonably reliable, but imperfect, test is employed to help ascertain the machine's condition.

This chapter assumes you are familiar with how TreeAge Pro converts influence diagrams into trees, as discussed in Chapters 3 and 25. See Chapter 23 for background on Bayes' revision.

Setting up a single forecast

First, create the chance nodes representing the true condition (whether the machine is faulty) and the test/forecast of that condition.

- Create a chance node called `Machine Condition`, with two outcomes Faulty and Not Faulty.
- Create a second chance node called `Test Result` with two outcomes: Test Positive and Test Negative.
- Draw an arc from Machine Condition to Test Result.

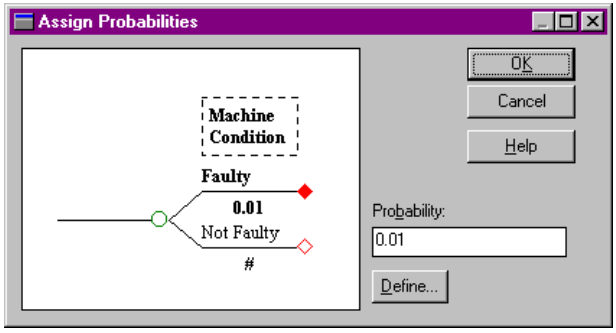


The direction of the arc indicates the direction of influence; probabilities of testing positive or negative are conditioned on the actual status of the machine. This conditioning reflects the data to which one typically has access, namely, the quality of the test.

The order in which the events unfold is opposite to the direction of influence. Even though Machine Condition influences Test Result, the outcomes of the test are known before the true machine condition is determined. Only by adding an intervening decision, as you will do eventually, can the model determine that the conversion order is opposite to the arc flow.

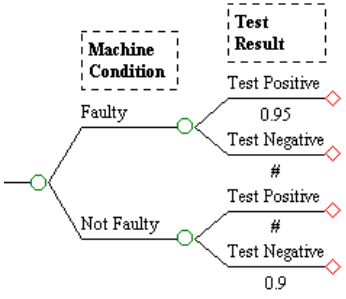
First, assign the probabilities to the defined events.

- ▷ Select the Machine Condition node, and choose **Diagram >** Probabilities… (or right-click and choose Probabilities…).
- ▷ Assign `0.01` to the Faulty outcome, and # to the Not Faulty outcome.



These values are called the prior probabilities; they represent the prevalence of this kind of fault in machines of this type. Next, enter the test node probabilities.

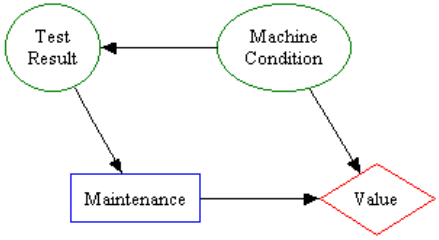
- ▷ Select the Test Result node, and choose **Diagram >** Probabilities….
- ▷ For the test outcomes conditional on Faulty, enter `0.95` for Test Positive, and # for Test Negative.
- ▷ For the test outcomes conditional on Not Faulty, enter # for Test Positive, and a value of `0.9` for Test Negative.



Next, as noted earlier, it will be necessary to add a decision based on the test results. Finally, a value node will be added.

Without an intervening decision, Bayes' revision will not be applied, and the chance nodes will not be ordered correctly. The value node is required to create a properly structured influence diagram.

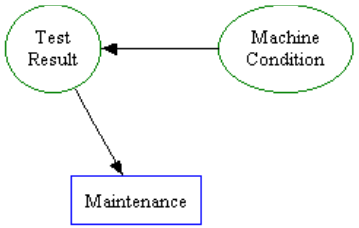
- ▷ Create a decision node called `Maintenance`, with two alternatives: Replace and Don't Replace.
- ▷ Create an arc from Test Result to Maintenance to indicate that the test results are known prior to making the decision.
- ▷ Create a value node called `Value`.
- ▷ Create two arcs, one from Maintenance to Value and the other from Machine Condition to Value.



Since the purpose of this example is to work through the steps needed for Bayes' revision, you can skip the process of assigning values at the value node. If, however, you wish to view the calculated probabilities in the converted tree, prior to doing the conversion you might indicate that the value node has a formula of 0 or 1. This will at least allow TreeAge Pro to perform roll back.

Understanding the structure

For the moment, ignore the value node, as it has no bearing on the conversion order. Focus your attention on the relationships among the other three nodes.



The only arcs are from Machine Condition to Test Result, and from Test Result to Maintenance.

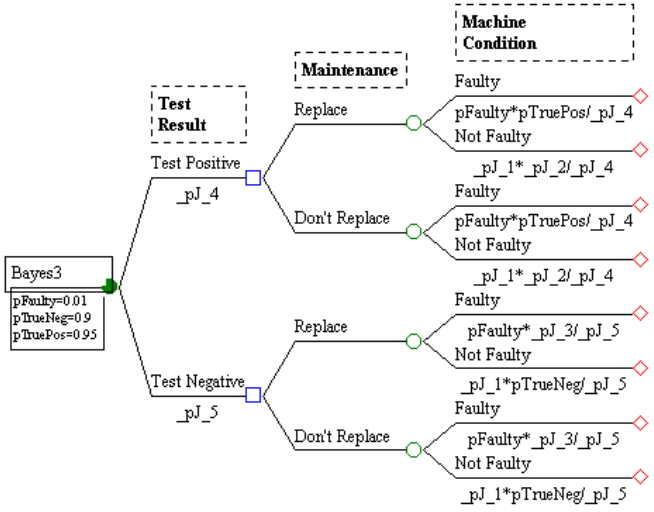
The absence of an arc from Machine Condition to Maintenance indicates that the uncertainty will not be resolved until *after* the decision is made. In contrast, the arc from Test Result to Maintenance specifies that the test result is known *before* the decision is made.

When TreeAge Pro converts your influence diagram into a tree, the nodes will be converted in the following order: Test Result, Maintenance, Machine Condition. This makes sense – the test result is first learned, then a decision is made, and then the true condition is learned.

You may be wondering about the arc between Machine Condition and Test Result, which indicates probabilistic dependence (described above). Why will these nodes be converted in the *opposite* order from the direction of arc flow? You may recall that node ordering is performed by first ordering the decisions, then grouping the chance nodes according to which decisions they precede and which they follow, and finally ordering the nodes within specific chance groups. Because Test Result and Machine Condition fall into different chance groups (one before and the other after the decision), the arc between them is not used to determine their relative ordering.

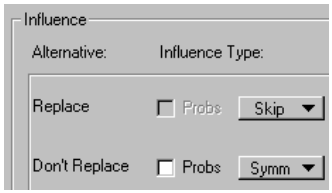
In cases such as this one – where the order of conversion will be opposite to the arc flow between two chance nodes – TreeAge Pro will automatically apply Bayes' revision when the influence diagram is converted into a tree, as illustrated below.

> ***Tip***:
> In the example file Bayes ID, prior and likelihood probabilities are assigned using variables, such as pTruePos. If you use variables, as is generally recommended, you will be able to perform sensitivity analysis on these values after converting the influence diagram to a tree. See Chapter 25 for more detail on using variables in the Assign Probabilities window.



Asymmetry inside the Bayesian model

In the example, one of the alternatives is costly but risk free. If the potentially faulty machine is replaced, the new machine will run the same regardless of whether the machine was actually faulty to begin with. In essence, the Machine Condition node no longer influences the Value node if Replace is chosen.



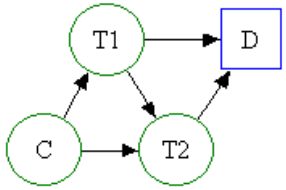
In this situation, you may add a structure-only arc from Maintenance to Machine Condition. You should edit the arc information to ensure it contains no probabilistic influence (i.e., the *original* condition of the machine is not dependent on the later decision). This arc will contain only structural influence. Now, you can indicate using the symmetry pop-up menu that the Machine Condition node should be *skipped* when the alternative Replace is chosen.

Converting the influence diagram into a tree (or viewing the enumerated values at the Value node) will allow you to view the resulting asymmetry.

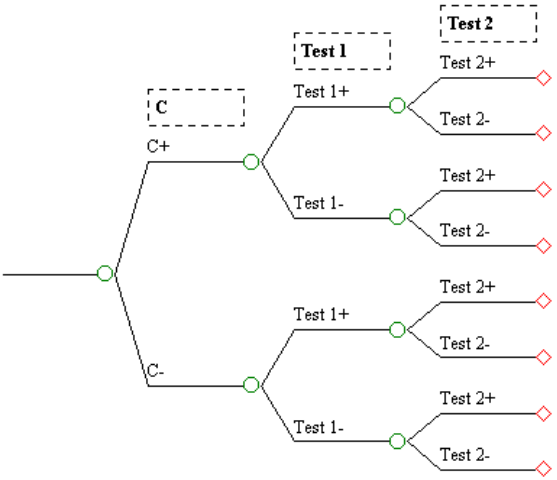
Another type of situation where asymmetry may affect the Bayes' revision structure is if there is an earlier decision whether or not to utilize the test.

Bayes' revision with sequential tests

TreeAge Pro knows how to apply Bayes' revision when more than one test (or predictor) is used on a single event (or hypothesis). Setting up the required influence diagram structure, shown below, is relatively simple.



The hard part comes next. In order for TreeAge Pro to be able to perform its probability revision calculations, you must provide all the information identified in the tree fragment.



For example, you must know the probability of testing positive on test 2, given that the underlying hypothesis C is positive and that test 1 returned a negative result. Obtaining these conditional probabilities is required.

It is possible to specify asymmetry in the arc from Test1 to Test2 if, for instance, Test2 should not be used if Test1 is negative.

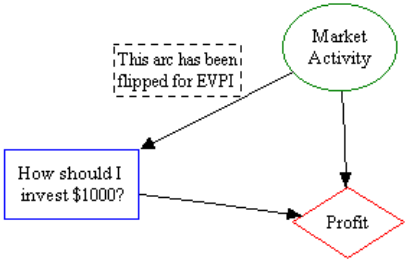
NOTE: Sequential test for different conditions require separate decision nodes after each condition.

Expected value of perfect information

The expected value of perfect information (EVPI), as described in Chapter 5, is a measure of the maximum amount one should be willing to pay for a predictor of an uncertain outcome or variable. See the discussion in that chapter for general information on the concept of EVPI.

EVPI can be calculated in the tree structure by inverting the time order of a chance node and a decision node. Under normal circumstances, the outcome of the chance node is not known before the decision is made. For EVPI calculations, we assume that the outcome is known before the decision.

To calculate EVPI, you should flip the arc pointing from the decision node to the chance node. Or, if there is no arc, add a new one from the chance node to the decision node. This will ensure that the uncertainty is resolved before the decision. The expected value of perfect information is equal to the value of this converted tree minus the value of the original tree.



A meaningful value of EVPI requires that there be arcs from both the decision node and the chance node to the final value node.

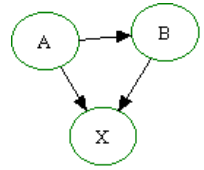
Revising the influence diagram in this way may result in some nodes becoming unnecessary. For instance, if there is an imperfect predictor of the chance node in question, it will become irrelevant in the presence of a perfect predictor. TreeAge Pro will not automatically remove these nodes for you, but they will not affect calculations in the converted tree.

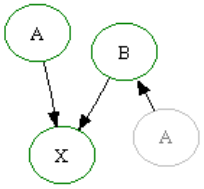
Other points to bear in mind:

- the application of Bayes' revision may be affected by these changes to the model; and
- any asymmetry specified in the original arc (from the decision node to the chance node) will be lost when the arc is flipped.

Clones

A clone of an influence diagram node is drawn in gray. Any changes you make to the clone will actually be made in the original ("master") node. Any arcs to or from the clone node act as if they pointed to or from the master node. Thus, the two influence diagrams in the left margin are functionally identical.



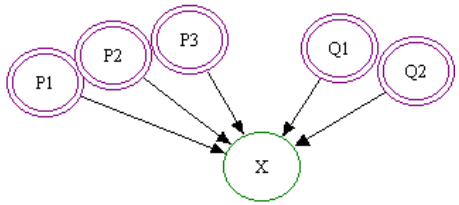


To create a clone, select a node and choose **Edit >** Create Clone. You may also destroy all clones of a node by selecting **Edit >** Destroy Clones when the clone master is selected.

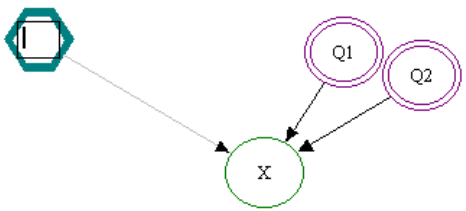
Note that the position of a clone is not relevant during conversion to tree. If graphical positioning is needed to resolve ambiguities in time-ordering, only the position of the master node will be considered.

Sub-models

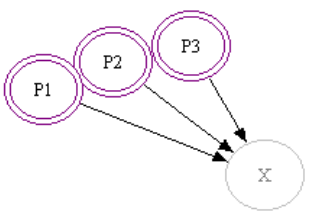
If you have a number of nodes which belong to the same logical group, you may find it helpful to put them into a sub-model. This can greatly simplify the printed and screen display of your model.



In the diagram above, the deterministic nodes are logically grouped. If you select the three "P" nodes and choose **Display >** Collapse to Sub-Model, the selected nodes will be hidden, and a single, blank hexagonal node created in its place.



The new node represents the *sub-model*; it is possible to name the sub-model as you wish. Even though a sub-model is displayed as a node in the diagram, it has no functionality of its own, except as an organizational tool. Double-clicking on the sub-model node will open a new document window displaying the collapsed nodes.



Sub-models and clones

TreeAge Pro will automatically create a clone of any node influencing, or influenced by, the sub-model nodes (in this case, X). This clone will be placed in the sub-model. You may also place new clones created in the main influence diagram window inside the sub-model. If there are sub-models in an influence diagram window when you create a clone, TreeAge Pro will offer to place the clone inside one of the sub-models instead of in the main window.

The position of a sub-model node is not relevant during conversion to tree. If graphical positioning is needed to resolve ambiguities in time-ordering, only the positions of the nodes it contains will be considered.

A sub-model may contain further sub-models. This recursive nesting can be dangerous while building a model, as it is easy to lose track of node locations. Even though the functionality is available, it may be unwise to place sub-models inside sub-models.

The sub-model is not a stand-alone document; it cannot be saved independently of the main model. On the other hand, it is possible to print a sub-model separately. The preference settings for the main influence diagram document apply to all of its sub-models.

TreeAge Pro 2007 Excel/COM Module

Part VI (Chapters 27–30)

Working with the Excel/COM Module

The chapters in Part VI provide details on working with the TreeAge Pro add-in for Excel, and an introduction to using TreeAgeProLib for scripting TreeAge Pro from macros (or other COM automation clients).

In this part:

- Chapter 27, "Linking Spreadsheets and Trees Using the Excel Module"
- Chapter 28, "Graphing and Reporting with the Excel Module
- Chapter 29, "Converting Spreadsheets to Influence Diagrams"
- Chapter 30, "Using the TreeAge Pro 2007 COM Automation Library"

CHAPTER 27

LINKING SPREADSHEETS AND TREES USING THE EXCEL/COM MODULE

This chapter describes how, with the TreeAge Pro Excel/COM module, the TreeAge add-in menu can be used to quickly set up and easily manage dynamic links from within Excel.

Chapter 20 covers dynamic linking, a powerful tool for integrating a TreeAge Pro decision tree and a Microsoft® Excel™ spreadsheet.

Chapter 30 covers an even more powerful method for exchanging information between trees and spreadsheets — the Excel/COM module's TreeAgeProLib scripting interface.

In this chapter:

- Linking a variable to a cell's value 356
- Setting up dynamic, bi-directional links 357
- "Linking" a spreadsheet to node calculations 361

Linking a variable to a cell's value

Chapter 20 covers the basic methods used to link tree variables to spreadsheet cell values. The process described there requires a series of steps: 1) name the cell in Excel; 2) copy the cell; 3) switch to the tree window; 4) create a variable; and 5) paste the link. If you are creating links to more than just a few cells, this switching back and forth between TreeAge Pro and Excel can be very time consuming.

Using the TreeAge Pro Excel module, however, it is possible to eliminate and combine many of the steps involved in setting up link outputs. This is also the case with setting up link inputs from tree variables used during dynamic, bi-directional linking, as will be described later in this chapter.

Macros targeting the TreeAgeProLib:

Starting with v2007, the TreeAge Pro Excel/COM module provides a powerful scripting interface that can be used to analyze and make changes to trees via Excel macros (or from other COM-compliant scripting/programming tools). The TreeAge Pro 2007 scripting interface ("TreeAgeProLib") is described in Chapter 30, and fully documented at the TreeAge support website.

Most of the linking commands in the TreeAge add-in menu are macros utilizing the TreeAgeProLib interface to communicate with and update your tree.

► **To create an output link from a cell to the active tree:**

▷ First, make sure the spreadsheet is saved (the following steps will record in the tree the location of the spreadsheet).

▷ With the source cell selected, pull down the **TreeAge** menu in Excel and choose Define Link Output (Excel to TreeAge).

▷ If the cell is not already named, you will be prompted for a name (one word, like a TreeAge Pro variable). Enter an optional description for the link.

▷ To define a tree variable using the link output, either select an existing variable from the list or enter a new variable name to create in the tree. Otherwise, the link that is created can be referenced in the tree using the Bilink(*n*) function.

▷ If you define a variable, you can choose to define it at either the root node or the currently selected node in the tree.

To link additional tree variables to other cells, simply repeat the steps above.

► **To see a list of links between the spreadsheet and the active tree:**

▷ Choosing **TreeAge >** Show Referenced Cells command will report the output (and input) links between the active spreadsheet and the active tree.

► **To select a tree node linked to a spreadsheet cell:**

▷ Choose **TreeAge >** Select Linked Tree Node with an output cell selected in order to locate a node with a variable, payoff, or probability using the link output.

Switch to the tree to see the selected node.

Setting up dynamic, bi-directional links

As explained in Chapter 20, if you have a complex set of calculations in a spreadsheet, rather than recreating the formula in a tree it may be possible to dynamically link the tree to the spreadsheet. With dynamic links, TreeAge Pro inputs a set of variables into corresponding spreadsheet cells, recalculates the spreadsheet, and returns cell values to the tree.

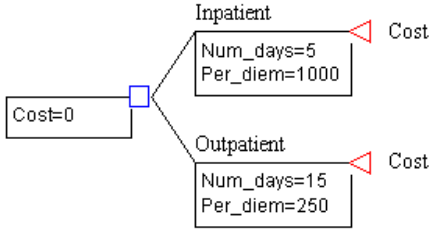
Chapter 20 included an illustrated example showing one method for setting up dynamic links between a tree and a spreadsheet. In that example, without using the Excel module it was necessary to switch back and forth between the spreadsheet and tree numerous times. With the Excel add-in's TreeAge menu, all of the steps can be completed in the spreadsheet.

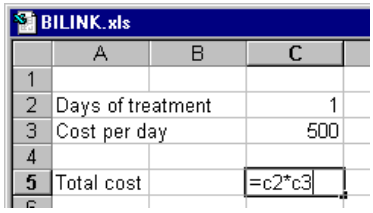
There are three steps to setting up the dynamic link:

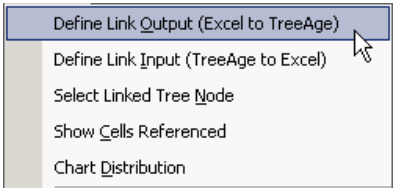
- link the tree to at least one *output* cell (or range of cells);
- assign one or more tree variable *inputs* to cells in the spreadsheet; and
- define a payoff variable or other parameter in the tree to use the spreadsheet output.

As in the example in Chapter 20, a dynamic link between a trivial tree and the simple spreadsheet calculator will be set up. Here, a simplified approach can be used which takes advantage of the TreeAge menu commands in Excel.

The tree and spreadsheet files used are Bilink2.tre and BILINK2.xls, found in TreeAge Pro's Tutorial Examples\Excel\ subdirectory.





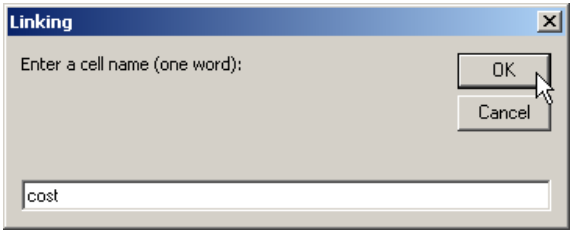

Define Link Output (Excel to TreeAge)
Define Link Input (TreeAge to Excel)
Select Linked Tree Node
Show Cells Referenced
Chart Distribution

► **To create the dynamic link output:**

▷ Save the spreadsheet.

▷ Select a calculation cell, and choose **TreeAge >** Define Link Output (Excel to TreeAge).

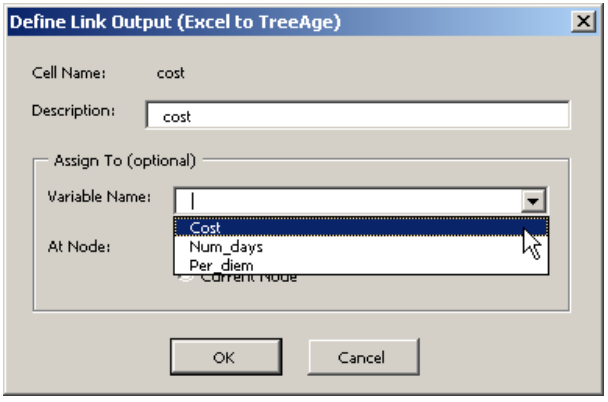
▷ If the cell is not named, you will be prompted for a name.



Tip:

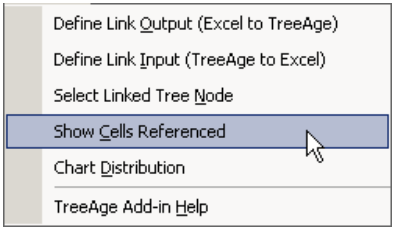
Another way to assign names to cells in the Excel spreadsheet is by selecting the cell (or range of cells) and choosing Insert > Name > Define…. The dialog displayed can be used to modify or delete existing cell names in the spreadsheet, if you make a mistake when initially specifying the name.

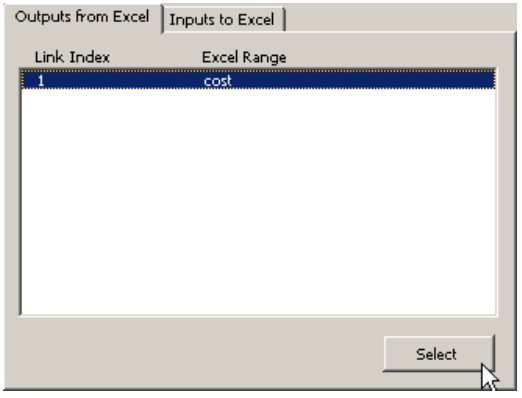
- Select an existing variable name or enter a new variable name to receive the link output, if desired, and choose to define it at either the root node or the currently selected node. (In this case, the variable Cost already exists in the tree.)



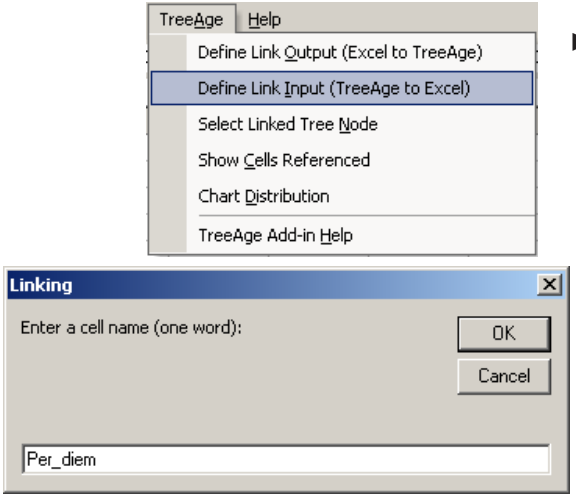
To create more than one output link, these steps can simply be repeated.

To see a list of cell outputs to the active tree, choose **TreeAge >** Show Cells Referenced and look at the dialog's Outputs from Excel tab.


Define Link Output (Excel to TreeAge)
Define Link Input (TreeAge to Excel)
Select Linked Tree Node
Show Cells Referenced
Chart Distribution
TreeAge Add-in Help



The next step in creating the dynamic link is to connect a variable to an input cell in the spreadsheet. This is what makes a link *dynamic*. To get the Cost output cell to return different values for each terminal node in the tree, the spreadsheet will use the Num_days and Per_diem tree variables as inputs to the Cost output calculation.

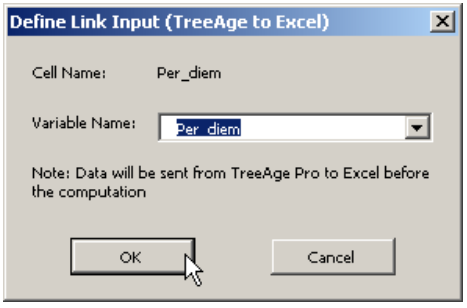


► **To create links from input variables to spreadsheet cells:**

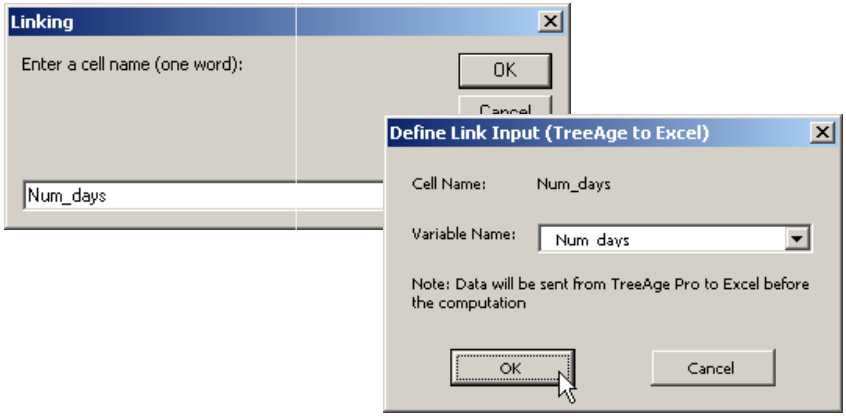
▷ Select a cell to receive the variable value, and choose **TreeAge >** Define Link Input (TreeAge to Excel).

▷ If the cell is not already named, you will be prompted for a name (one word, like a TreeAge Pro variable).

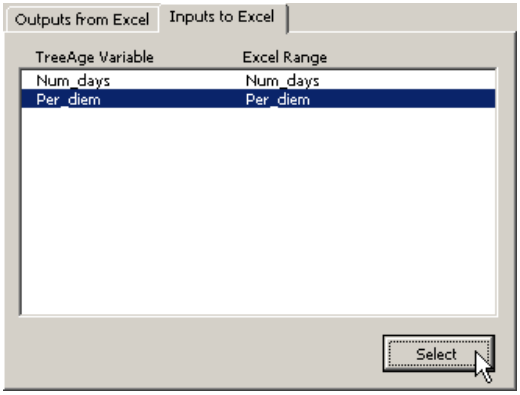
▷ In the Define Link Input dialog, select a variable from the list which will input its value to the spreadsheet cell.



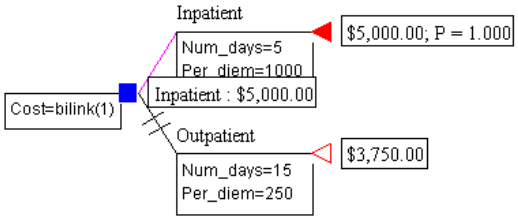
Repeat these steps for each input variable.



To see a list of connected variable–input cell pairs, choose **TreeAge >** Show Cells Referenced and switch to the Inputs to Excel tab. Clicking on an item in the list and clicking the Select button will move to that cell in the spreadsheet.



Once you have set up both the input and output linkages, try rolling back your tree.



To *change* which variable a cell is paired with, or vice versa, switch to the tree window and delete the input link using TreeAge Pro's **Edit >** Links > Dynamic Links… command. Then recreate the desired pairing, either from TreeAge Pro or Excel. See Chapter 20 for more details on working with TreeAge Pro's Dynamic Links dialog.

"Linking" a spreadsheet to node calculations

The use of DDE to link a spreadsheet to calculated node values, in particular expected values, was discussed in Chapter 20. Among the limitations of this approach to ***linking to trees***, however, is that it is not controlled by the target document (i.e., the spreadsheet). While the value of a DDE link is updated automatically when both documents are open and the tree is recalculated, there is no easy way from within Excel to have the DDE link force the recalculation of the tree.

This section describes two more dynamic methods for retrieving values from a tree and using them in a spreadsheet.

Writing macros targeting the TreeAgeProLib

Starting with v2007, the TreeAge Pro Excel/COM module provides a powerful scripting interface that can be used to analyze and make changes to trees via Excel macros (or from other COM-compliant scripting/programming tools). The TreeAge Pro 2007 scripting interface ("TreeAgeProLib") is described in Chapter 30, and fully documented at the TreeAge support website.

The TreeCalc( ) worksheet function described below is a simple macro application utilizing the TreeAgeProLib interface.

> ***Note:***
> The TreeCalc() worksheet function should not be used directly or indirectly in the calculation of a dynamic link output cell, as it will result in errors.

TreeCalc( ) worksheet function

The TreeAge add-in adds a special worksheet function to Excel, called TreeCalc( ), which can be a useful tool for retrieving node calculations from the active tree.

► **To use a tree node calculation in a cell formula:**

▷ In the cell formula, type the expression `=TreeCalc("expr")`, where the *expr* (must be in quotes) is a valid calculation in the active tree.

(NOTE: Excel worksheet functions generally do not recalculate unless one of their arguments references a cell that has been changed. You can force a spreadsheet to recalculate using the F9 key, or another macro.)

The expression will be calculated at the root node if no second argument is provided, or if you enter a second argument that evaluates to 1. If the second argument evaluates to 2, the calculation will instead be performed at the currently selected node.

So, the cell formulas:

```
=TreeCalc("probX")        or
=TreeCalc("probX",1)
```

will return the value of the variable probX at the root node of the active tree. To evaluate probX at the currently selected node, enter:

```
=TreeCalc("probX", 2)
```

The expression calculated using the TreeCalc() function can be any formula that might be used in the active tree, including a TreeAge Pro function. If the tree includes, for example, a sampling distribution with the index 1, the formula:

```
=TreeCalc("Dist(1;1)")
```

will force TreeAge Pro to randomly sample a value from the distribution and return that value to the cell.

"Behind the function"

Here is an abbreviated version of the add-in macro that runs when a cell references the TreeCalc( ) worksheet function.

```
Public Function TreeCalc(expr As String, _
                         where as integer) As String
    Dim app As New TreeAgeProLib.ApplicationObj
    Dim tree As TreeAgeProLib.TreeObj
    Dim nod As TreeAgeProLib.NodeObj
    Set tree = app.getTreeObj("", "") ' was CurrentTree
    node = tree.GetNodeObj()
    TreeCalc = node.Evaluate(expr)
End Function
```

This code excerpt highlights some basic aspects of the TreeCalc( ) function. For example, it simply calculates the expression using the open tree (if there is one).

As described in Chapter 30, you can write your own more complex VBA macros (or scripts in other ActiveX client applications/environments) that use script steps like those shown here to connect to TreeAge Pro, open or access a tree, and modify or analyze the document.

Chapter 28

Graphing, Reporting, and Modeling using Excel

This chapter includes details on creating Excel charts and reports from TreeAge Pro using the Excel/COM module.

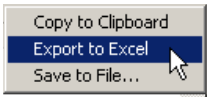
In addition, it describes the use of the Excel add-in to add or update distributions, tables, and variables used in your models.

In this chapter:

- Exporting TreeAge Pro text reports 364
- Creating Excel charts 364
- Editing variables, tables, and distributions in Excel 367

Exporting TreeAge Pro text reports

As described in Chapter 6, the contents of any text report dialog displayed in TreeAge Pro can be copied to the clipboard or saved to a text file. With the Excel module, you have an additional option: to have TreeAge Pro automatically copy the text report into a new Excel worksheet.



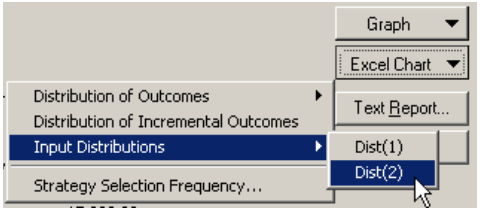
If an existing worksheet is found with the default name, TreeAge will prompt you to reuse the same worksheet for the new report.

Creating Excel charts

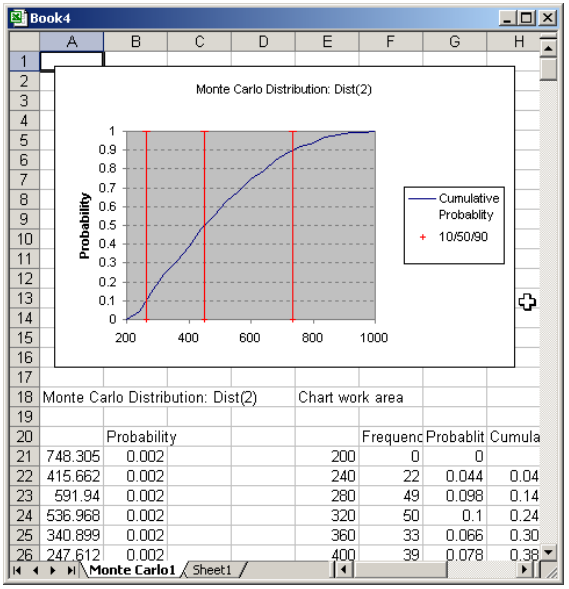
As described in Chapter 6, there are many kinds of changes that can be made to TreeAge Pro graphs, including modifications to: the contents, location, and style of labels; the axis scales and numeric format; and the size and location of the entire graph area. To perform customization beyond TreeAge Pro's capabilities, it is sometimes possible to export the graph drawing objects as a Windows metafile (*.wmf file). Modifying the metafile object's parts can be time consuming and may not be a feasible approach for more involved tasks.

Another option, if you have the Excel module, is to create an Excel chart object using the TreeAge Pro add-in for Excel. Chart objects in Excel can be manipulated and customized to a greater extent than is possible in the TreeAge Pro graphs. Charts can be resized and reshaped, and text elements, axes, and legends can be modified and reformatted. Excel charts can also use chart subtypes that do not exist in TreeAge Pro, such as stacked bars. Charts in Excel also have the advantage of being easy to copy into other Microsoft® Office™ documents.

For example, open any tree and perform a Monte Carlo simulation at the root node. If you have the Excel add-in, the simulation output window will include both a regular Graph menu button, and an Excel Chart menu button.



If you select a graph type from the menu, TreeAge Pro will export the data to a new spreadsheet and create an appropriate Excel chart to represent the data. TreeAge Pro will also start Excel, if necessary, which may take a few seconds.



Excel's standard set of charting tools can be used to customize the resulting chart. When the chart is selected, the TreeAge add-in menu will disappear and the Excel Chart tool bar will appear.



Chart commands can also be accessed by right-clicking on elements of the chart.

Converting existing TreeAge graphs into Excel charts

Like the Monte Carlo Simulation output window, the Markov Cohort Analysis and Cost-Effectiveness Sensitivity Analysis output windows also include an Excel Chart menu button duplicating the regular Graph menu button.

Most TreeAge Pro graphs also include an **Excel Chart** button (graphs that cannot be represented as Excel charts include region graphs, such as 2-way sensitivity analyses). This allows you to convert an open TreeAge graph into an Excel chart.

Customizing distribution charts with the Excel add-in

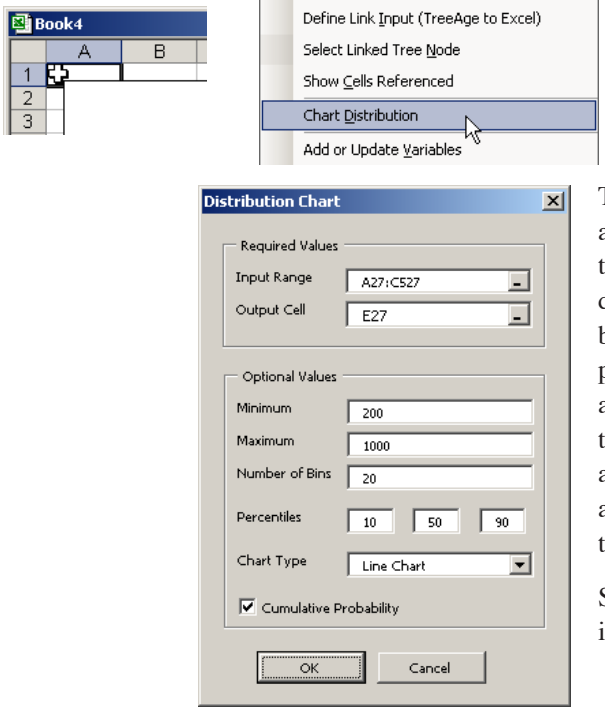
Whereas the initial style used to display distribution graphs in TreeAge Pro is as a non-cumulative histogram (bar chart), a distribution graph created from TreeAge Pro's Excel Graph menu will be displayed initially as a cumulative line graph (see the example above). Another difference is the addition of lines marking the 10/50/90 percentiles.

As with all other Excel charts, the TreeAge distribution graphs can be customized in a variety of ways using the standard charting tools. However, some aspects of a distribution graph cannot be easily customized using the charting tools, including: non-cumulative format; percentile lines; bar/histogram format; number of intervals. The TreeAge Pro add-in includes a Distribution Chart dialog for handling such changes.

► **To customize a distribution chart created by TreeAge Pro:**

▷ In a distribution chart worksheet created by TreeAge Pro, select any cell (rather than the chart).

▷ Choose **TreeAge >** Distribution Chart.

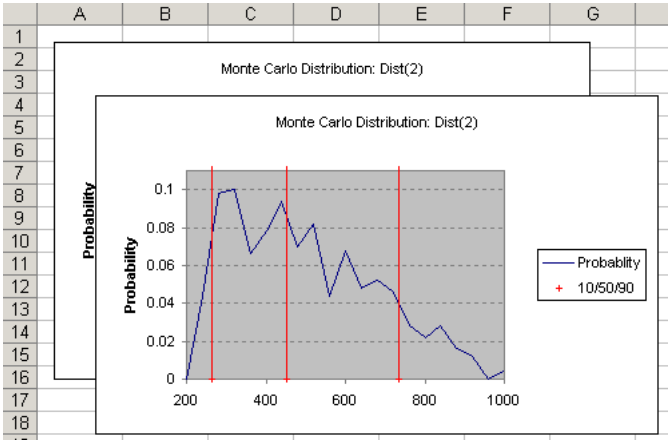


This will open a dialog which allows you to create a copy of the distribution chart with modified settings. Use this dialog to, for example, display a distribution chart in non-cumulative format, as a bar chart instead of a line graph, or with different percentile lines. Making changes to these settings and clicking OK will create a new Excel chart, on top of any existing charts that are present. Clicking and dragging the new chart to a new location will allow you to see the old charts, and delete or move them if necessary.

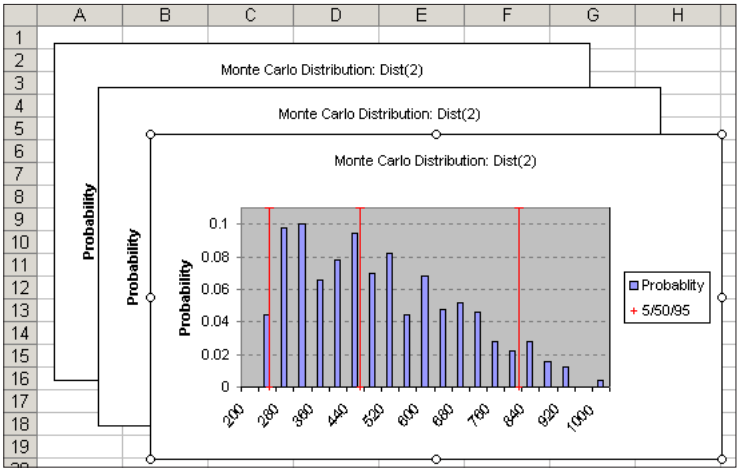
Some customizations are illustrated on the following page.



The chart can be recreated as a non cumulative line graph, for example. Click and drag the new graph to a different location to compare it to the original graph.



A bar chart/histogram can be created, and the marked percentiles can be changed (to 5/50/95 in the example below).



The Distribution Chart dialog can also be used to create a distribution graph from a new set of data. This allows you to use the TreeAge Pro add-in for Excel to easily create distribution graphs. The source data range is specified in the Distribution Chart dialog, and should have a column for the data point values, and a column for the probability of each data point. The Distribution Chart dialog can then be used to group the data points into intervals, and graph these interval's probabilities.

The settings from the Distribution Chart dialog are saved with the spreadsheet, making it easy to change the chart format later.

Editing variables, tables, and distributions in Excel

The Variables and Tables dialog is used to manage your tree's lists of variables (Chapter 13) and tables (Chapter 18). To manage the tree's list of distributions, the Distributions dialog is used instead (Chapter 17).

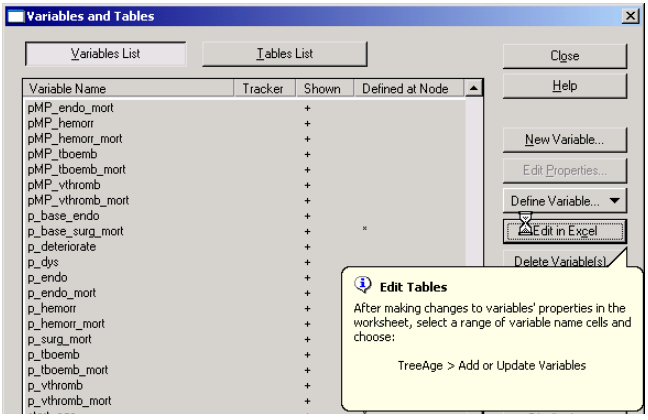
In both TreeAge Pro Suite and TreeAge Pro Excel, the Variables and Tables dialog and the Distribution dialog include an **Edit in Excel…** button. If you have the Excel module, use this feature in order to efficiently edit the properties (and definitions) of variables, tables, and distributions.

Exporting lists of variables

Once a tree includes at least one variable, it is easy to start editing/adding variables in Excel.

► **To edit variables using Excel:**

▷ Open a tree. In the Variables and Tables dialog, select any number of variables and click the **Edit in Excel** button.



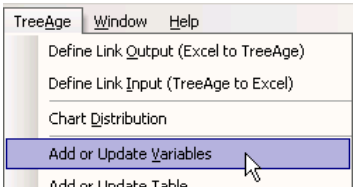
TreeAge Pro will create a new worksheet called "tree_variables" and construct a table showing the selected variables, their properties (e.g., description, sensitivity range), and their definitions from the root node.

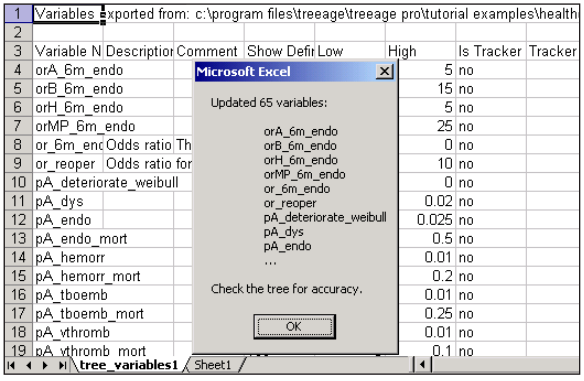
Editing and creating variables

After making changes to the variables' properties, or adding rows to the table for new variables, use the TreeAge add-in menu to update the active tree.

▷ In Excel, select a range of variable names to update/add. (To update/add *all* variables in the worksheet list, select cell A1, in the top row.)

▷ Choose **TreeAge >** Add or Update Variables. The list of updated variables will be confirmed.





Notes on editing variables in Excel:

- Starting with v2007, changing a variable name in the worksheet and then updating **will** change the original variable's name in the tree, and update the model. If you want to create a new variable instead, delete the "Original Variable Name" column's value, and updating will instead treat it as a new, different variable.
- The list of variables is not linked to a particular tree, which means the Add or Update Variables command can be used to add the list of variables to a different tree.
- Some changes are not allowed. In particular, a tracker cannot be changed to a regular variable. Only make changes to the "Is Tracker" field when adding a new variable.

Editing and creating tables

The process used to edit or create TreeAge Pro tables in Excel is somewhat like that described for variables.

► **To edit TreeAge Pro tables using Excel:**

▷ Open a tree and choose **Values >** Variables and Tables. Activate the Tables List and select any number of tables.

▷ Click **Edit in Excel**.

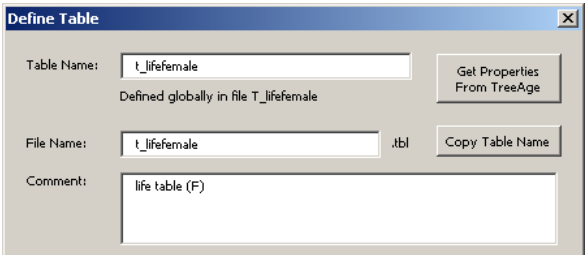


A new worksheet will be created for each table, based on its name. Each worksheet displays the corresponding table's properties (e.g., name, comment, lookup method), column headings, and values.

After making changes to a table in its worksheet, use the TreeAge add-in menu to send the updated information to the active tree.

▷ In Excel, select a cell anywhere in the table that you want to update. Then, choose **TreeAge >** Add or Update Table.

▷ Excel will display a Define Table dialog in which you can confirm the changes before creating/updating the table in TreeAge.



New tables can also be created from Excel, in either of two ways.

► **To create a table using Excel:**

▷ Export an existing table from TreeAge Pro, change its name and other properties in the worksheet, and then use the Add or Update Table command; or

▷ Select a table of indexes and values in any worksheet and choose **TreeAge >** Add or Update Table command.

▷ Excel will display a Table Properties dialog.

Some notes regarding editing tables in Excel:

- Changing a table name in the worksheet does not change the original table's name in the tree. Instead, it creates a new copy of the table with the specified name.
- Both table files as well as tables stored in a package can be edited using this procedure.
- If the changes are made to tables from a package, the Add or Update Variables add-in menu command can be used to add the tables to a different tree.
- New tables created using the Add or Update Tables command are initially stored as table files, even if a package file is open in TreeAge. If the table is referenced by a tree, and the tree is later saved as a package, a copy of the table will be included with the package.
- Some changes are not allowed. In particular, a global table cannot be changed to a packaged table from Excel.

Editing and creating distributions

The process used to edit or create TreeAge Pro distributions in Excel is very similar to that described for variables.

► **To edit TreeAge Pro distributions using Excel:**

▷ Open a tree and choose **Values >** Distributions. Select any number of distributions.

▷ Click **Edit in Excel**.



TreeAge Pro will create a new worksheet called "tree_distributions" and construct a table showing the selected distributions and their properties (e.g., index, name, parameters). After making changes to the distributions' properties, or adding rows to the table for new distributions, use the TreeAge add-in menu to update the active tree.

▷ In Excel, select a range of distributions to update/add. (To update/add *all* distributions in the worksheet list, select cell A1, in the top row.)

▷ Choose **TreeAge >** Add or Update Distributions. The list of updated distributions will be confirmed.

Some notes regarding editing tables in Excel:

- Changing a distribution index in the worksheet does not change the original distribution's index in the tree. Instead, it updates or creates a different distribution with the specified index.
- The list of distributions is not linked to a particular tree, which means the Add or Update Distributions command can be used to add the distributions to a different tree.

CHAPTER 29

IMPORTING INFLUENCE DIAGRAMS FROM SPREADSHEETS

This chapter shows how, using the Excel module, an existing Microsoft® Excel™ spreadsheet can be converted into a TreeAge Pro influence diagram, including links to the numeric input cells and output calculation cells in the spreadsheet.

Chapters 20 and 29 describe the use of dynamic links between a spreadsheet and a tree.

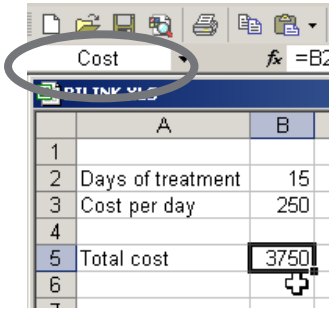
See Chapters 3, 25, and 26 for details on building and customizing influence diagrams, as well as converting them to decision trees.

In this chapter:

- Importing a simple spreadsheet 372
- Cleaning up imported influence diagrams..... 381

Importing a simple spreadsheet

The TreeAge Pro Excel module includes a powerful feature for creating an influence diagram from a spreadsheet. The resulting influence diagram is dynamically linked to the spreadsheet, which means that the TreeAge Pro model can be structurally modified, converted into a decision tree, and calculated using any of TreeAge Pro's analytical tools. Analyses will update input cells in the spreadsheet (e.g., interest_rate and principal) and then use the dynamically recalculated values from output cells (e.g., interest, profit) in the tree's calculations.



Setting up the spreadsheet

The first step is to set up the spreadsheet so that it can be interpreted by TreeAge Pro.

The basic requirement is that you name any output cells or input cells that you want to include in the influence diagram. Furthermore, for an accurate representation of your spreadsheet's formulas and dependencies, you should then use the cell names that you create in the relevant cell formulas (instead of A1-style references to cells).

► **To assign a name to a cell in Excel:**

▷ Select a cell.

▷ Click in the Name box (shown at left) on the Excel toolbar. Type a one word name (like a TreeAge variable name)

▷ Press the ENTER key to apply the name to the cell.

If a named cell is subsequently selected, the name will appear in the Name box (instead of the A1-style cell reference). The cell name should be used in formulas referencing the cell, in place of the A1-style or R1C1-style cell references. If you use the mouse to select a named cell while typing a formula in another cell, Excel will insert the cell name in the formula.

> ***Tip***:
> The other way to assign names to cells in the Excel spreadsheet is by selecting the cell (or range of cells) and choosing Insert > Name > Define…. The dialog displayed is used to manage existing names in the spreadsheet.

Converting the spreadsheet

Once an input or output cell has been named, TreeAge Pro can convert the spreadsheet's formulas into a influence diagram. The tutorial uses a simple spreadsheet called Product.xls, found in the TreeAge Pro's Tutorial Examples\Excel\ subdirectory.

Product.xls

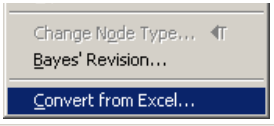
	B	C	D	E	F	G
3	**Inputs**				**Results**	
4						
5	Price		$250		Revenue	1250000
6	Units sold		5,000		Cost	800000
7						
8	Fixed costs		$500,000		Profit	450000
9	Variable costs		$300,000			
10						

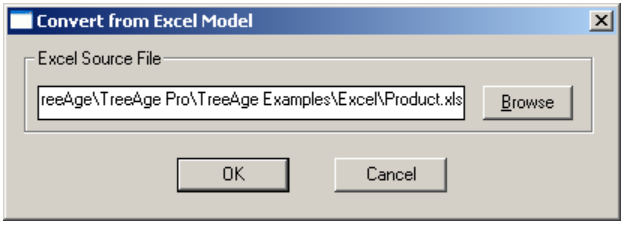
This Product spreadsheet already uses cell names for all inputs and outputs.

► **To convert a spreadsheet into an influence diagram:**

▷ First, save a copy of the source spreadsheet for TreeAge Pro to use. (Don't use the original unless you are familiar with the changes that TreeAge Pro will make in the spreadsheet during calculations). Note where the spreadsheet is saved.

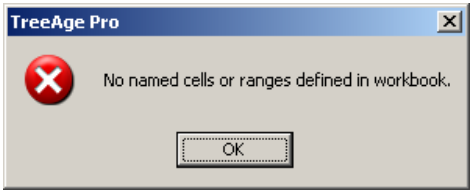
▷ In TreeAge Pro, create a new, empty influence diagram. Choose **Diagram >** Convert from Excel…, and browse for the saved Excel spreadsheet.





▷ TreeAge Pro will start Excel and open the source spreadsheet, if necessary, and check for named cells in the spreadsheet.

If no named cells are found, you will get an error message.



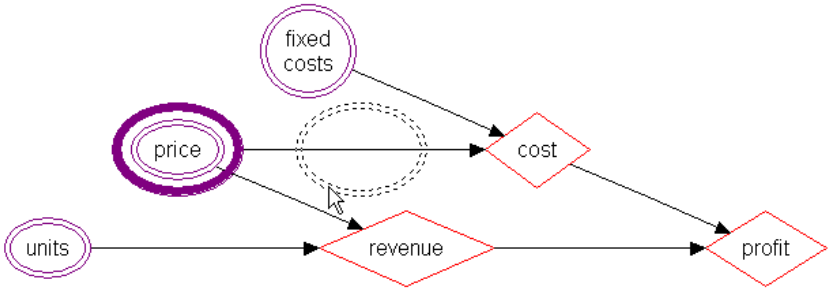
TreeAge Pro will determine which are inputs and which are outputs based on their contents/formulas.

A cell which references other cells in its formula will become a value node in the influence diagram; the value node will be dynamically linked to the output cell.

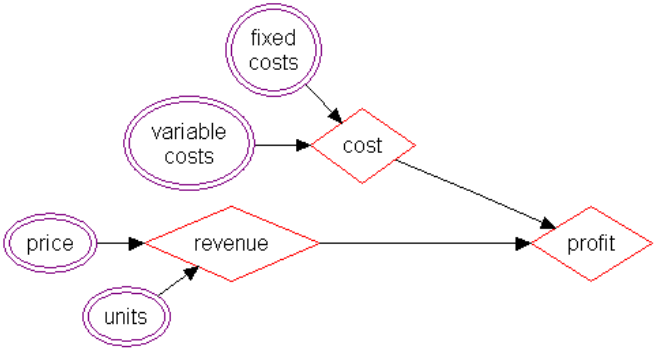
Numerically-defined cells, or cells whose formulas do not reference any other cells, will become deterministic nodes; each deterministic node is a variable in the influence diagram, and its value will exported to (linked with) the corresponding input cell.

Arcs will be created from a deterministic node to any value nodes that references it. Arcs will also be created from a value node to any other value nodes that reference it in the spreadsheet formula.

The initial result of converting the Product.xls spreadsheet is shown below. Placement of nodes in the influence diagram is based in part on the layout of the source cells in the spreadsheet. However, there will normally be some overlap of nodes and arcs.



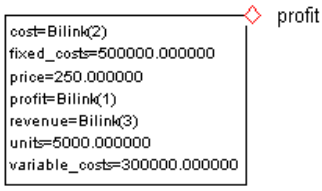
The standard influence diagram editing tools, described in Chapters 3 and 26, can be used to rearrange nodes and to curve arcs to achieve a clear layout.



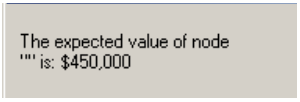
As noted above, influence diagrams created from spreadsheets are *deterministic*, initially. This simply means that they do not include any chance nodes (uncertainties), only deterministic nodes representing input variables and value nodes representing calculations of outcomes. It is very easy to convert deterministic nodes to chance nodes, however, in order to introduce uncertainty in the model.

First, however, try converting the influence diagram to a tree in order to calculate the model and see the dynamic links in action.

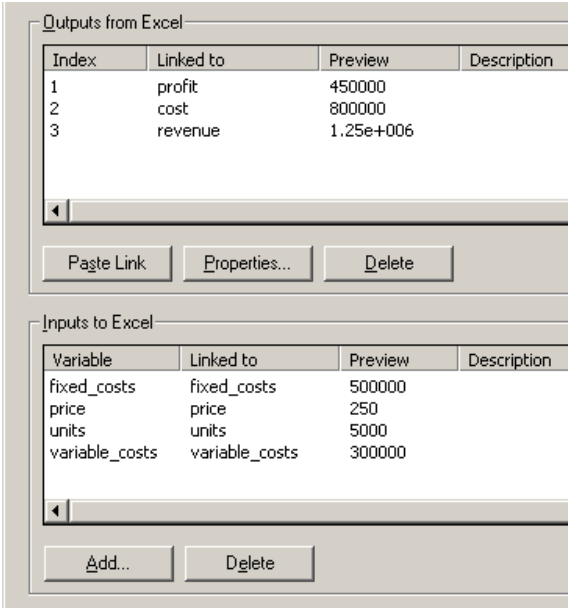
- In the influence diagram, choose **File >** Convert to Tree.



- Either roll back the resulting tree, or select the root node and choose **Analysis >** Expected Value.

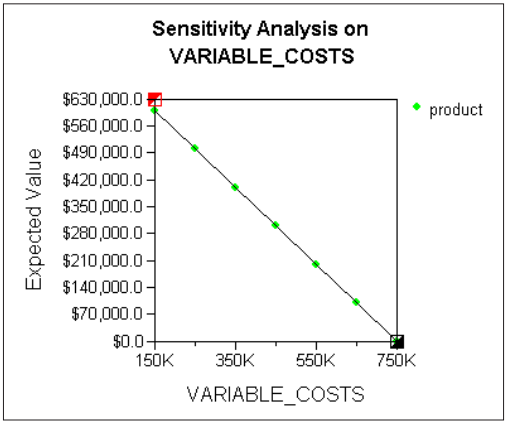


Profit is defined as the result of a dynamic link calculation. To see which cell Profit references, or to see a list of other cells used during the calculation of the tree, choose **Edit >** Links > Dynamic Links….



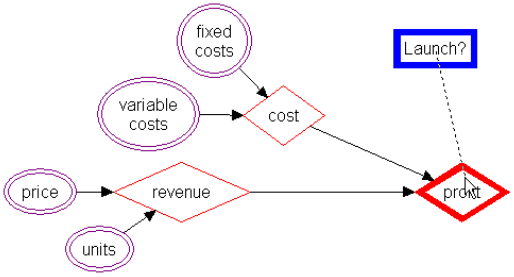
The same link information can be viewed or modified in the influence diagram *before* converting to the tree.

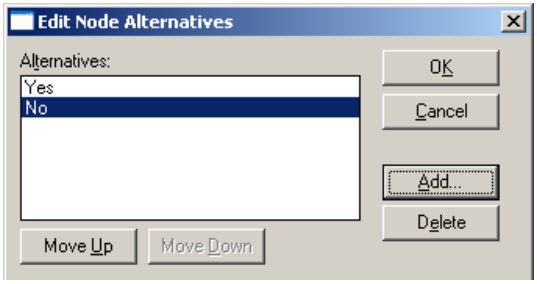
You also may wish to test the linkage by performing a sensitivity analysis on a variable that is an *input* to the spreadsheet's calculation of profit — for example, variable costs, from 150k to 750k.



Customizing the influence diagram

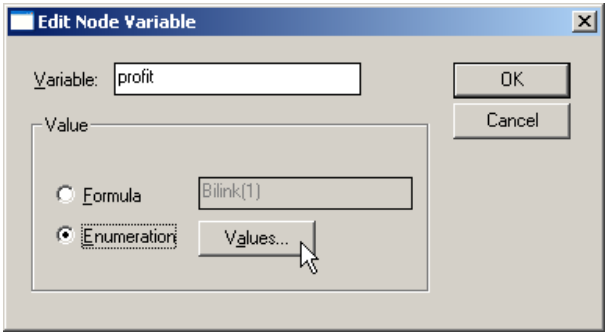
The logical next step is to introduce the two aspects of a decision analytic model that distinguish it from a standard spreadsheet model: decisions and uncertainties. First, add a decision representing whether or not to pursue the product launch in the first place.





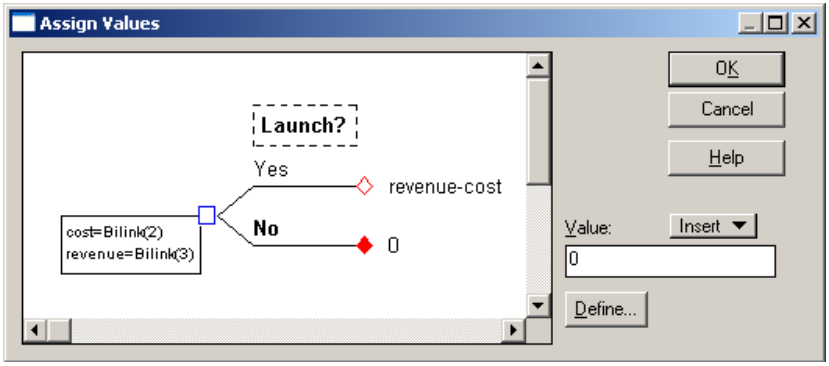
Double-click on the decision to add the two choices. Then draw an arc from the decision to the profit node, to indicate the influence of the decision on the value of the outcome. To specify influence, double-click on the profit node.

Profit is currently calculated using the Bilink(1) function, which links to the value of the Profit cell in the spreadsheet. To indicate that profit will differ depending on our action, select the Enumeration option and click the **Values…** button.



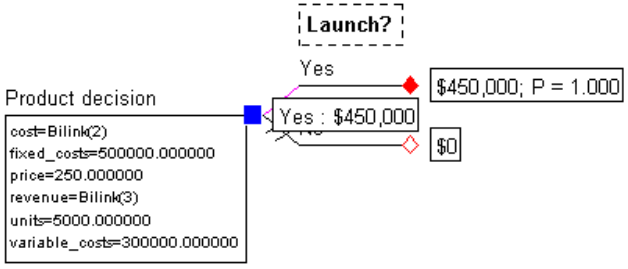
It is possible to continue using the Bilink(1) function to calculate profit when the decision is to launch. Or, as is illustrated on the following page, it is also possible to instead reproduce the formula from the profit cell (i.e., profit=revenue-cost) in the payoff of the launch option.

For the value of profit when the decision is made *not* to launch the product, simply enter 0.



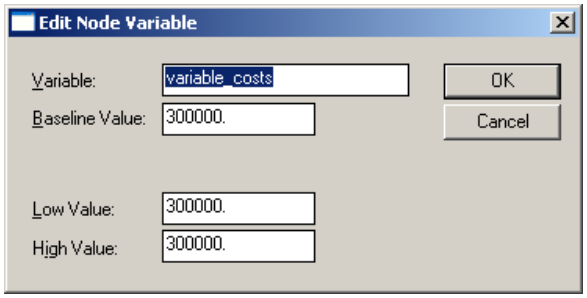
Note that, in the original tree, the variables linked to the cost and revenue cells in the spreadsheet were not actually utilized. These cells were used within the spreadsheet calculations, but only the profit cell was referenced in the tree calculations. Now, the profit cell will be ignored and an equivalent calculation done in the tree using revenue and cost.

Convert the influence diagram to a decision tree, check the tree's calculation method preferences to make sure it will report useful results, and then roll it back.



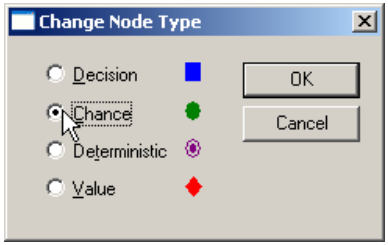
Earlier in the tutorial a sensitivity analysis was performed on variable costs, perhaps to examine uncertainty in the estimate of 300,000 used in the spreadsheet calculations. It is also possible to make this a structural uncertainty visible in the model, by changing the deterministic node to a chance node.

Return to the influence diagram. Double-click on the variable cost deterministic node to check the baseline value used (i.e., the cell value).



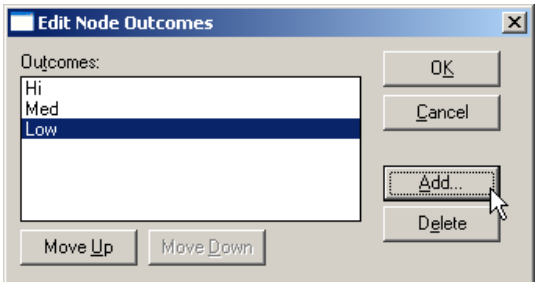
Close the Edit Node Variable dialog, and either click on the Change Node Type tool bar button, or choose **Diagram >** Change Node Type…. Change the variable costs node to a chance node.

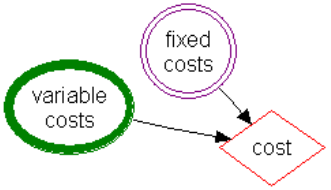




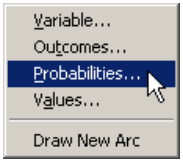
For a list of possible outcomes for the chance node, three outcomes can be used: high, medium, or low variable costs, with relative probabilities of 25%, 50%, and 25%, respectively.

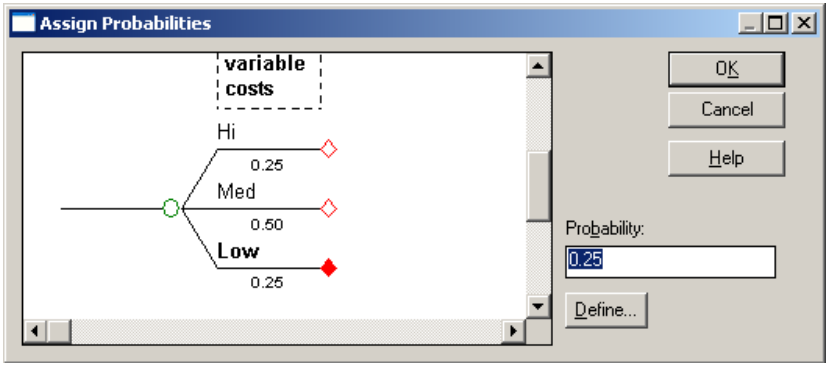
To assign the list of outcomes, double-click on the variable costs chance node.



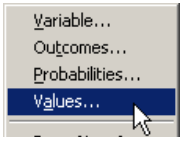


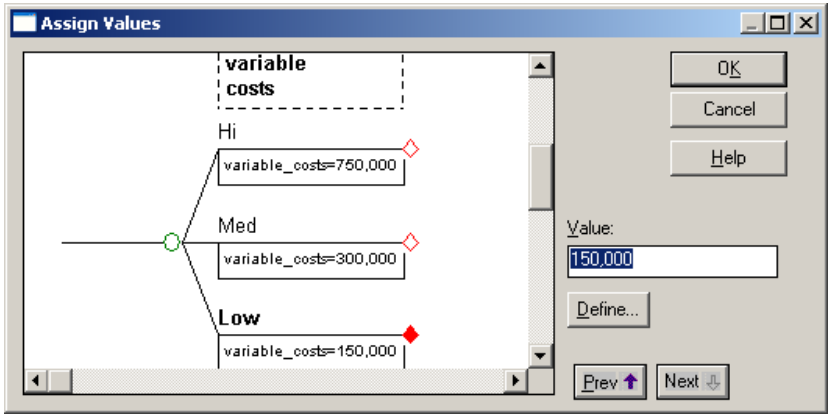
Close the Edit Node Outcomes dialog. To enter outcome probabilities, choose **Diagram >** Probabilities… (or right-click on the node).



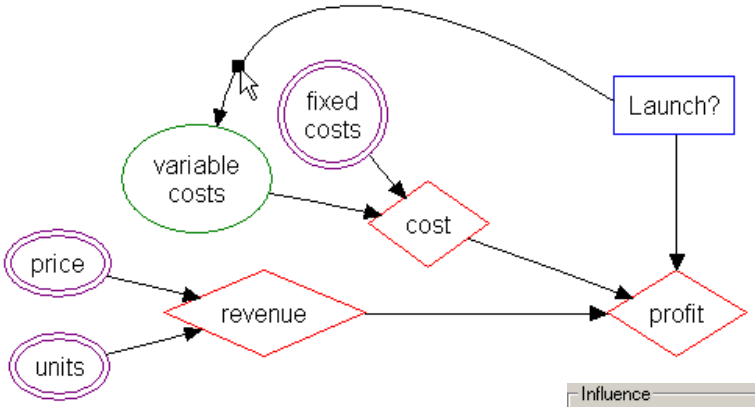


Close the Assign Probabilities dialog. To assign the different possible values of variable costs, choose **Diagram >** Values….

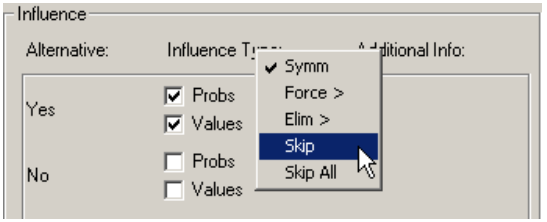




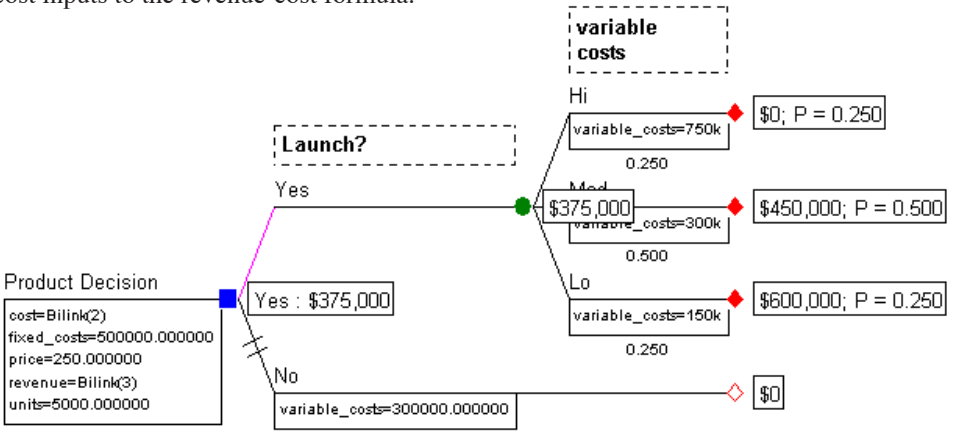
The last structural change that should be made is the addition of a structure-only arc from the decision node to the chance node, indicating that the variable costs uncertainty is irrelevant when the decision is to not launch the product.



All of the Probs and Values boxes can be unchecked, and the No option should be changed to Skip.

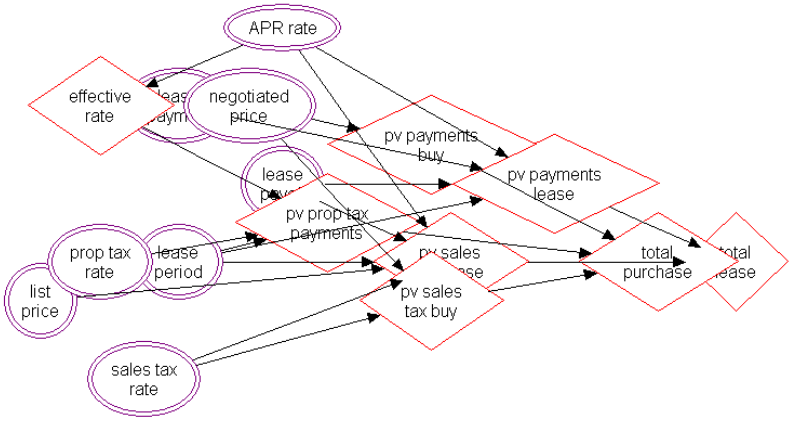


Choose **File >** Convert to Tree again. Confirm that the rolled back tree uses the dynamic link to calculate different payoff values for the different variable cost inputs to the revenue-cost formula.



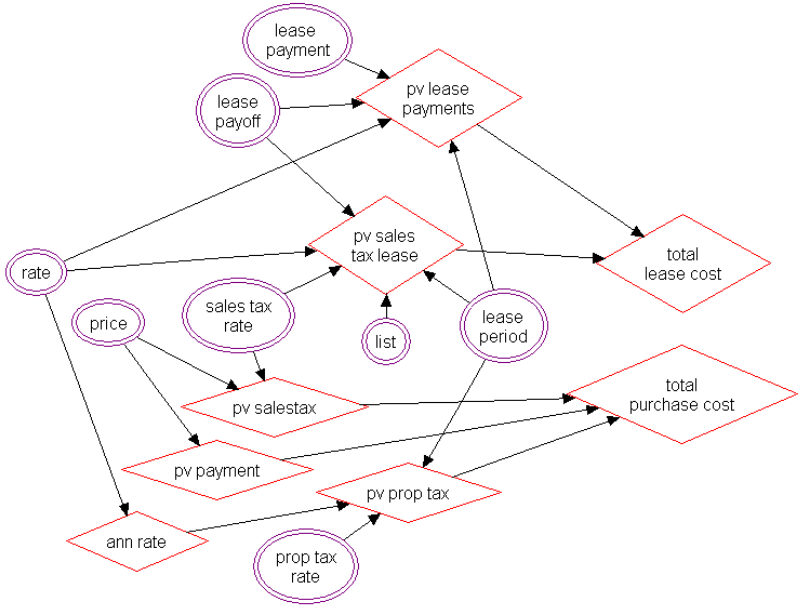
Cleaning up imported influence diagrams

As noted in the previous section, TreeAge Pro will convert every named cell in the spreadsheet into either a deterministic or a value node. In a spreadsheet with many named cells it may take some work to clean up the imported influence diagram.



In the example shown here, created from the Lease.xls spreadsheet, rearranging the nodes to clarify relationships takes some effort.

In this model, and more complex models, often some nodes can be removed. Since all calculations are initially contained in the spreadsheet, any node can be deleted (except final value nodes, which will not have arcs leading out).



Nodes that you do not care about presenting in the influence diagram, and that you do not plan to convert to chance nodes or to perform sensitivity analysis on, can be deleted

CHAPTER 30

USING THE TREEAGE PRO 2007 COM AUTOMATION LIBRARY

Starting with v2007, the TreeAge Pro Excel/COM module provides a powerful scripting interface that can be used to analyze and make changes to trees via Excel macros (or from other COM-compliant scripting/programming tools).

The scripting interface is very similar to the TreeAge Pro Interactive library, described in Appendix F. Both libraries are documented in detail at the TreeAge support website.

In this chapter:

- What is the "TreeAgeProLib" COM library? 384
- Using the TreeAgeProLib library in a macro 384

What is the "TreeAgeProLib" COM library?

Starting with v2007, users of the TreeAge Pro Excel module also have access to a full-featured COM automation interface for TreeAge Pro 2007.

What this means is that it a simple task for TreeAge Pro users to create custom Excel macros (or scripts in other COM automation clients, e.g., C++, Python, etc.) that can update and analyze trees and other documents in TreeAge Pro. Other applications with scripting interfaces can be used, for example Visual Basic or C++.

Utilizing your choice of Windows scripting or development tools, with Excel VBA being the most popular, you can control trees, graphs, and Monte Carlo output using a set of commands similar to those used with TreeAge Pro Interactive (see below).

A simple macro might, for example, open the tree in TreeAge Pro, make changes to selected values (probabilities, payoff components, etc.), and then perform a series of Monte Carlo simulations or Markov analyses.

The scripting interface can be used to automate repetitive tasks, for example:

- **Monte Carlo simulation** – manage a series of Monte Carlo simulations from Excel (or another application supporting COM scripting/macros)
- **Markov analyses** – manage a series of Markov cohort analyses from Excel

The TreeAge Pro 2007 library interface provides access to trees and other documents via straightforward methods and properties.

A sample listing of objects is provided here to illustrate the applications of the scripting interface.

TreeAge Pro – selected objects

ApplicationObj	*Controls access to TreeAge Pro documents*
TreeObj	*Wrapper around tree document*
NodeObj	*Wrapper for a single tree node*
CEOutput	*Output of a C/E analysis*
GraphObj	*Wrapper for a graph*
TextReportObj	*Provide access to output of analyses*
MarkovOutput	*Output of a Markov analysis*
MonteParams	*Parameters for a Monte Carlo simulation*
MonteOutput	*Output of a Monte Carlo simulation*

Each object has different properties and methods. Here are a few for TreeObj:

TreeObj – selected methods and properties

MonteOutput	*Output of a Monte Carlo simulation*
SelectRoot	*Selects the root node of the tree*
GetNodeObj	*Creates a NodeObj at the selected node*
Markov	*Runs a Markov analysis*
MonteCarlo	*Runs a Monte Carlo simulation*
CostEff	*Runs a baseline cost-effectiveness analysis*

Using the TreeAgeProLib library in a macro

A simple example macro illustrating usage of some common objects and methods is shown below.

```
'--- This VBA macro could be linked to button in spreadsheet.
Sub Analyze_CE_Baseline()
    Dim tree As TreeAgeProLib.TreeObj
    Dim nod As TreeAgeProLib.NodeObj
    Dim cea As TreeAgeProLib.CEOutput
    Dim graph As TreeAgeProLib.GraphObj
    Dim ok as integer

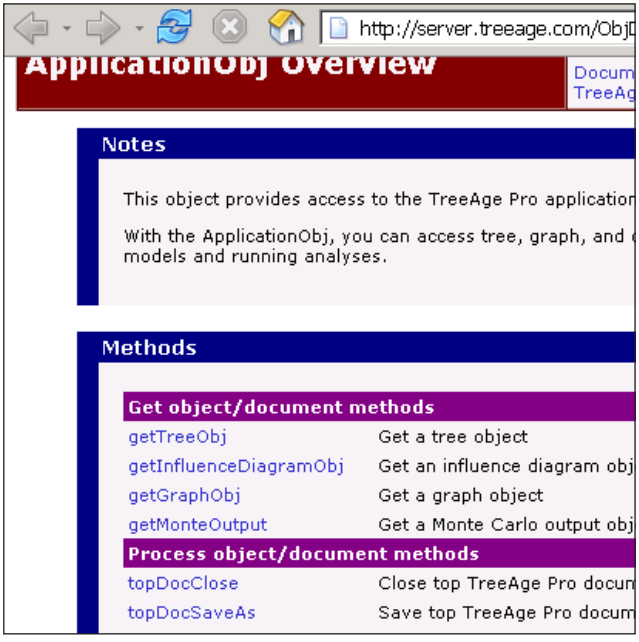
    Dim treeage As New TreeAgeProLib.ApplicationObj '--- connect
    Set tree = treeage.getTreeObj("", "") '-- empty string gets top tree
    If not tree is nothing Then           '-- if tree open ...
        tree.SelectRoot
        Set nod = tree.GetNodeObj
        If nod.type = nt_decision then    '-- if root is decision do CEA
            Set cea = tree.CostEff        '-- analysis will show in TreeAge Pro
            '--- If we want graph:  Set gObj = ceo.GetGraph(1, "")
            '--- Or, we could GetTextReport() and show in Excel
        End If
    End If
End Sub
'-------------------------------------------------------------
```

This section of VBA code is similar to the "Excel Sample.xls" for TreeAge Pro Interactive shown in Appendix E. The main difference between the code sample here, and the TreeAge Pro Interactive sample shown later in Appendix E, is that the TreeAge Pro code uses the ApplicationObj (TreeAge Pro Interactive on the other hand simply creates TreeObj objects directly).

Example macro projects, as well as complete object documentation, can be found at the TreeAge web site:

http://server.treeage.com/objDocs

A good introduction to the functionality of the interface can be gained by opening some of the example projects under TreeAge Pro\Tutorial Examples\ActiveX\ and accessing the Macro project editor (press Alt + F11).



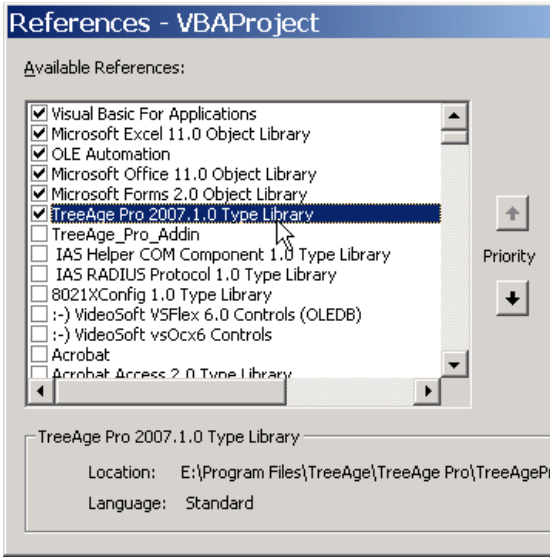
This page illustrates some of the tools available when working with the TreeAge Pro 2007 library in Excel's VBA macro window.

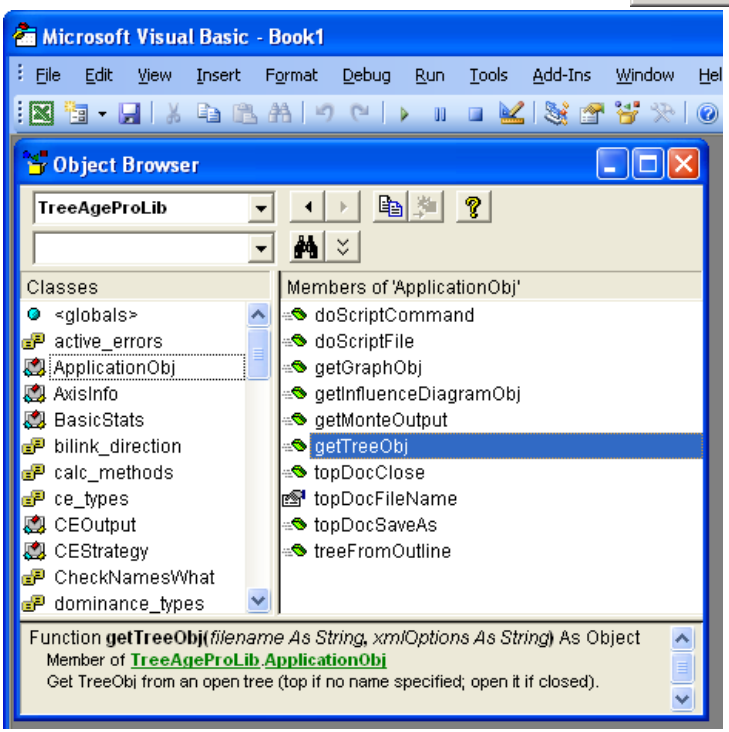
The first step in creating your own macro project using the TreeAge Pro 2007 library is to add a reference to the library.

To open the VBA Editor for Excel, press Alt+F11.

To add a reference to TreeAge Pro 2007, choose Tools > References.

You can then browse the available functionality by pressing F2 to open the Object Browser, and selecting the TreeAgeProLib from the list of referenced libraries.





TreeAge Pro 2007 Healthcare Module

Parts VII – VIII (Chapters 31–37)

Cost-Effectiveness Analysis with the Healthcare Module

The chapters in Part VII provide information on modifying existing decision trees for cost-effectiveness analysis. Cost-effectiveness analysis, sensitivity analysis, and Monte Carlo simulation are also covered in detail.

In this part:

- Chapter 31, "Building and Analyzing Cost-Effectiveness Models"
- Chapter 32, "Cost-Effectiveness Modeling and Analysis Options"
- Chapter 33, "Cost-Effectiveness Simulation Graphs and Reports"

CHAPTER 31

BUILDING AND ANALYZING COST-EFFECTIVENESS MODELS

Healthcare decisions often must take into account differences in both cost and effectiveness between competing treatments, technologies, or strategies. This chapter covers preparing an existing TreeAge Pro decision tree for cost-effectiveness calculations, and performing and interpreting baseline cost-effectiveness analysis.

Chapter 32 provides information on intermediate and advanced cost-effectiveness modeling and analysis options.

In this chapter:

- Before you begin 392
- Preparing a tree for cost-effectiveness calculations..... 393
- Performing cost-effectiveness analysis 398
- Dominance and incremental cost-effectiveness..... 400
- One-way cost-effectiveness sensitivity analysis..... 405

Before you begin

Cost-effectiveness analysis (CEA) is a collection of methods used by health economists and policy researchers to evaluate policy recommendations on the basis of two different attribute scales, cost and quality-adjusted life expectancy, for example.

The TreeAge Pro Healthcare module is designed to facilitate building and analyzing cost-effectiveness decision trees and Markov models — both so-called *early* models which rely heavily on rough guesses about outcomes and parameters, as well as *evidence-based* models that summarize extensive research and meta-analysis.

The basis of CEA in a decision tree is the calculation of expected values for each strategy at a decision node (refer to Part I of this manual for details on expected values). TreeAge Pro then creates a CEA table, in which the strategies are listed in order of increasing cost, and calculates incremental cost and effectiveness values for neighboring pairs of options. This is used to determine conditions of *dominance* and to calculate *incremental cost-effectiveness ratios* (ICERs), as described in this chapter.

For more background, users are strongly encouraged to consult some of the many references on medical decision making, for instance:

- Periodicals including *Medical Decision Making* (Sage Science Press) and *Value in Health* (Blackwell Publishing).
- *Decision Making in Health and Medicine,* Hunink, and Glasziou (2001), Cambridge University.
- *Cost-effectiveness in Health and Medicine*, Gold, Siegel, Russell, and Weinstein, eds. (1996), New York: Oxford Univ. Press.
- *Methods for the Economic Evaluation of Health Care Programmes, 3rd Ed.*, Drummond, et al (2005), New York: Oxford Univ. Press:
- *Designing and Conducting Cost-Effectiveness Analyses in Medicine and Health Care*, Muennig. (1996), New York: Oxford Univ. Press.
- *Meta-Analysis, Decision Analysis, and Cost-Effectiveness Analysis*, Petitti, (1994), New York: Oxford Univ. Press.
- *Medical Decision Making*, Sox, et al. (1988), Boston: Butterworth-Heinemann.

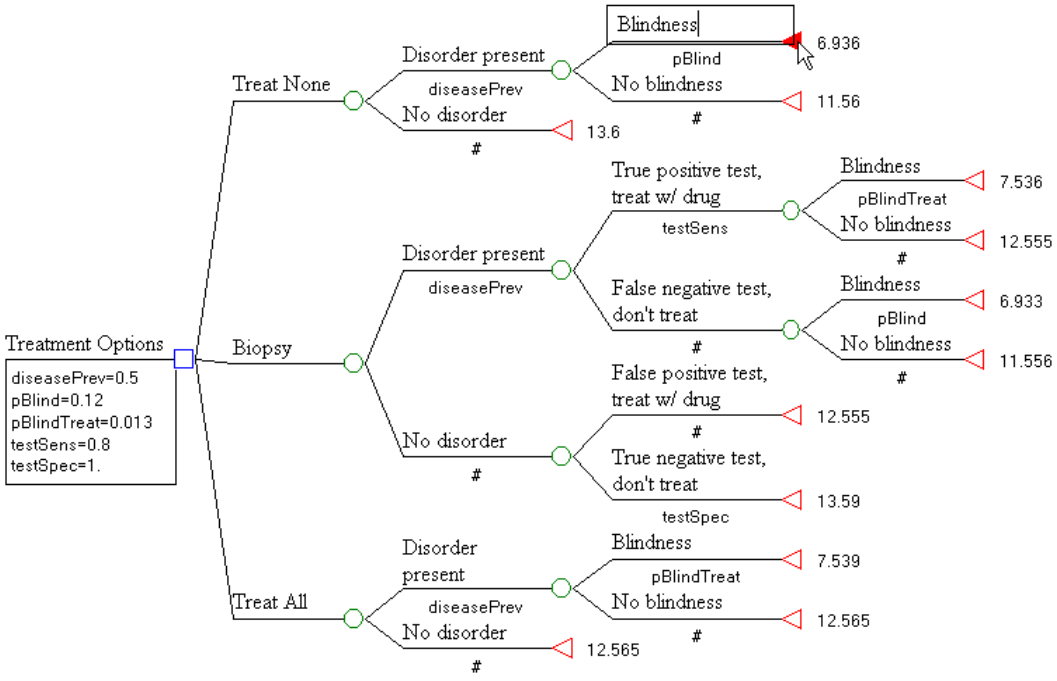
Preparing a tree for cost-effectiveness calculations

This chapter uses an example decision tree that already has costs and effectiveness values in two payoffs. The tree is an adaptation of a model described in other decision analysis tutorials.

▷ Open the example file Blindness Prevention, from TreeAge Pro's Tutorial Examples\Healthcare subdirectory.

The model deals with a hypothetical population presenting clinical signs of a possible, but not certain, early-stage autoimmune disorder. If the condition is present, and if it progresses, blindness will result. An imperfect test (biopsy, with the possibility of false negatives) can help determine whether an individual has the disorder.

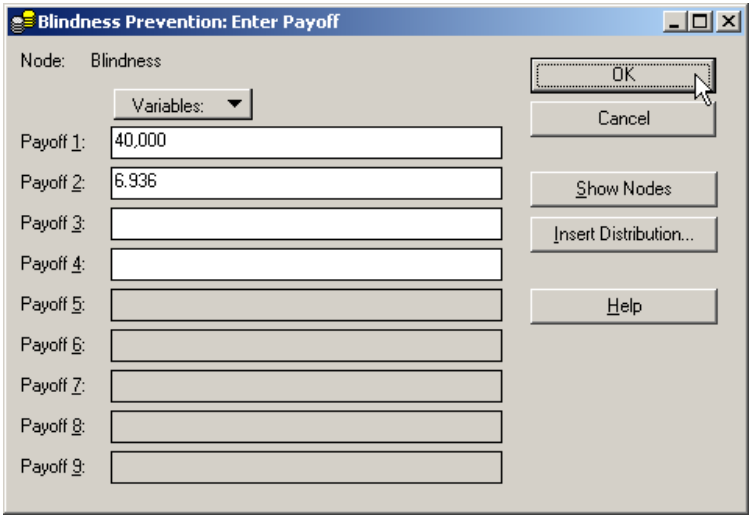
The model presumes that a reasonably effective and inexpensive therapy exists which lowers the probability of progression to blindness. To keep the example tree small, the possibility of side effects of the therapy are not modeled with a chance node, like the other uncertainties. The side effects are instead already factored into the costs and life expectancy at the end of each path including treatment.



The Cost-Effectiveness calculation method

Before making any changes to the Blindness Prevention tree, take a moment to examine the assignment of payoffs at a terminal node.

▷ Double-click on the Blindness terminal node in the Treat None subtree (indicated by an arrow in the picture on the previous page). This will open the Enter Payoff dialog for that node.



The numeric effectiveness payoffs displayed on the face of the tree, measuring quality-adjusted life expectancy (QALYs), are entered in payoff 2. In addition, every terminal node also has a numeric cost payoff assigned in payoff 1. The cost payoffs are not displayed, however, because the tree's calculation preferences are set to Simple calculations, using only payoff #2.

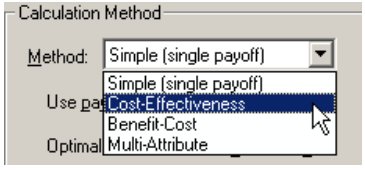
Note on regional (European) numeric settings:

If your computer is set up to use commas (",") to represent decimals, rather than periods ("."), you should enter numbers in TreeAge Pro in this fashion, just as you would in a spreadsheet or calculator (even in example trees where numbers already appear using period decimals). If you need to setup a tree for use under different regional numeric settings, however, TreeAge Pro also has preference settings that enable a tree to override a computer's regional settings or reverse the usage of decimals; see Chapter 12 for details.

Before you can perform cost-effectiveness analysis, the tree's calculation method preferences must be set up correctly.

► **To prepare a tree for cost-effectiveness calculations:**

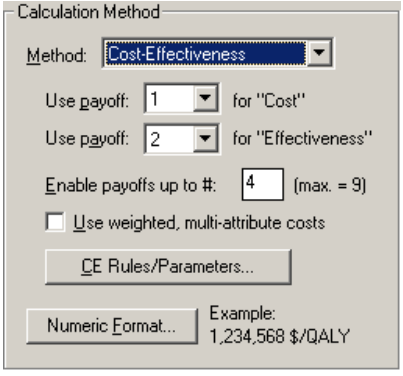
▷ Close the Enter Payoff window.



▷ Choose **Edit >** Preferences…. The default category is the Calculation Method preference page.

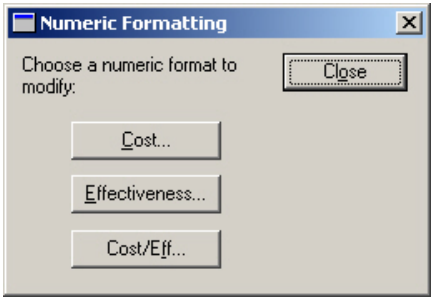
▷ Change the Method selection from Simple to **Cost-Effectiveness**.

This will enable the special analyses and reports described in this and following chapters. Note that the tree can be changed back to Simple calculations at any time.



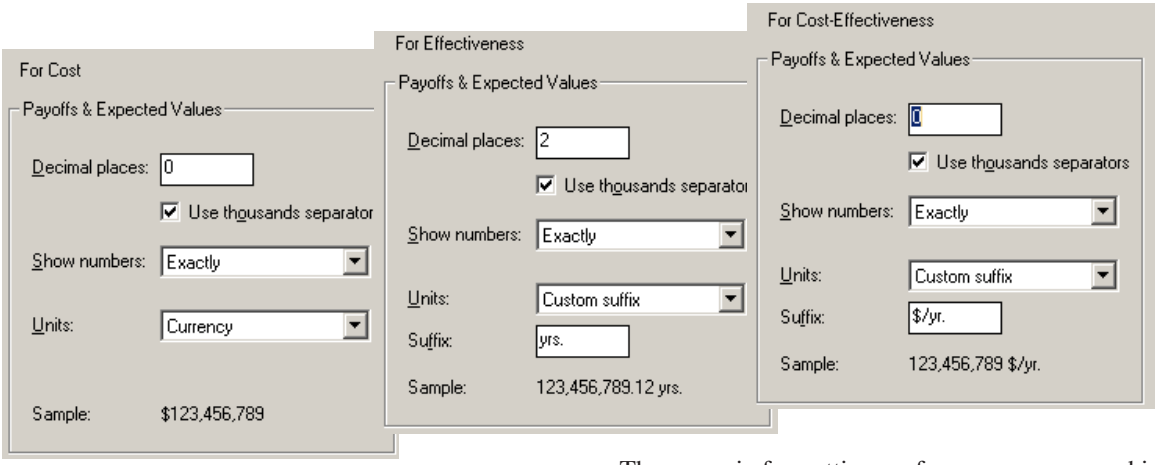
▷ By default, the Cost-Effectiveness calculation method uses payoff 1 for costs, and payoff 2 for effectiveness. This can be changed. However, in the Blindness Prevention tree, costs are already in payoff 1 and effectiveness values are in payoff 2.

▷ To set the numeric formatting for calculated costs, effectiveness values, and CE ratios, click the **Numeric Format…** button.



The tree stores separate numeric formatting preferences for cost and effectiveness (as well as the other 7 payoff sets), and another group of settings for ratios.

The graphic at the top of the next page illustrates some possible settings for these three sets of numeric formatting used in cost-effectiveness analysis.



The numeric formatting preferences are covered in greater detail in Chapters 2 and 5.

If you plan to create more cost-effectiveness trees, you can use this tree's calculation method, numeric formatting, and other preferences as the defaults when creating new trees.

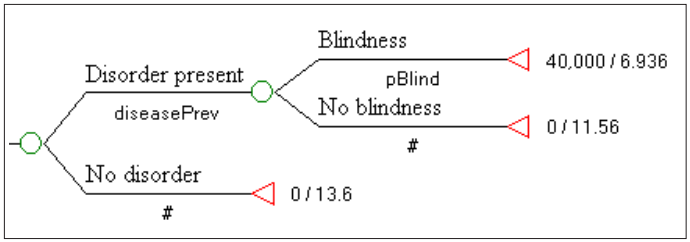
► **To make the current preferences the defaults for new trees:**

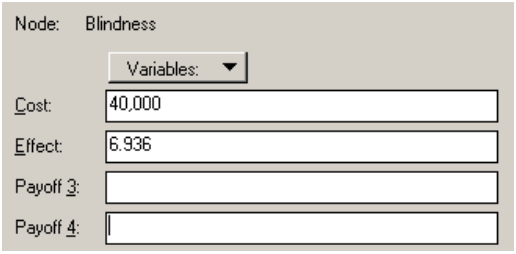
▷ In the Preferences dialog, check the box labeled Save settings as default.

▷ Press ENTER or click OK to close the Preferences dialog and return to the tree.

The next chapter covers additional, special cost-effectiveness preferences, for example: setting a threshold ICER for roll back; inverting effectiveness calculations (if lower values are preferred); or specifying costs using a weighted combination of multiple payoffs.

With the CE calculation method active, the tree displays both the cost and effectiveness payoff expressions for each visible terminal node. A forward slash ("/") sign is used to visually separate the two payoffs; it *does not* mean that cost will be divided by effectiveness during roll back or other tree calculations, however.





In addition, if you double-click on a terminal node to open an Enter Payoff window, the two active payoffs are labeled "Cost" and "Eff."

The tree is now ready for cost-effectiveness analysis. The remainder of this chapter will cover performing cost-effectiveness analysis using the TreeAge Pro Healthcare module. Chapter 32 covers additional cost-effectiveness modeling and analysis options .

Note:

Users of TreeAge Pro who *do not* have the Healthcare module can open trees using the Cost-Effectiveness calculation method, but cannot analyze them. It is possible to change the calculation method of such trees to Simple in order to analyze one of the payoffs, even without the Healthcare module. See the introduction to this manual for more information on the Healthcare module.

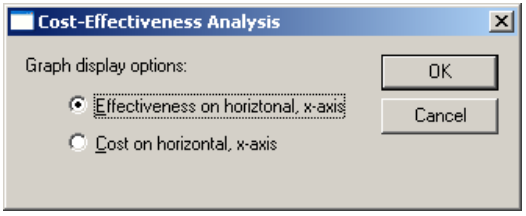
Performing cost-effectiveness analysis

TreeAge Pro's cost-effectiveness (CE) graph, and the text report underlying it, are the fundamental tools for cost-effectiveness analysis of your decision trees. They display the key information from the analysis, including incremental values and conditions of dominance.

► **To generate a CE graph:**

▷ Select the decision node, and choose **Analysis >** Cost-Effectiveness….

After finishing calculations, TreeAge Pro presents a dialog with two layout options. Because the CE graph plots each strategy as a point corresponding to its calculated cost and effectiveness, you can choose whether to display cost or effectiveness on the horizontal (X) axis. This selection does not affect calculations or the text report.



▷ Select the desired X axis option, then click OK or press ENTER.

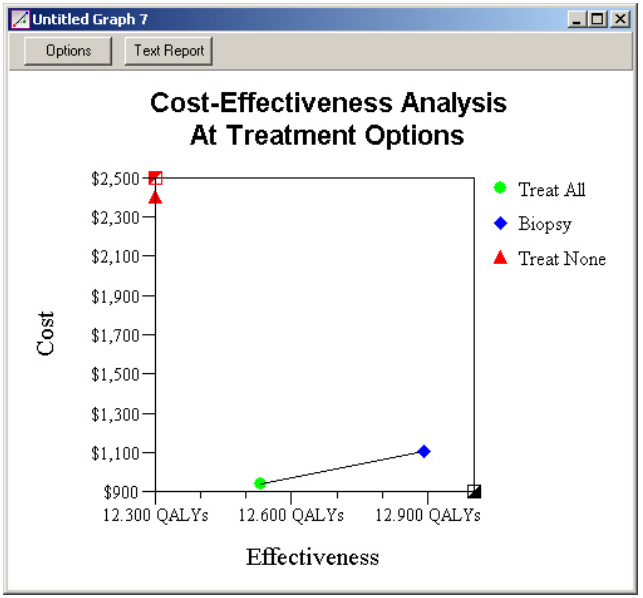
The process of interpreting the results of the cost-effectiveness analysis graphically is described in detail later in this chapter. A quick overview is given here.

If the graph includes multiple options which are not *dominated*, these are connected by line segments defining a *cost-effective frontier*, or set of possibly optimal choices. The lowest cost option is always part of this frontier; if it dominates all comparators, the graph will have no lines.

Details are provided later in this chapter on the rules used to determine which options are excluded from the cost-effective frontier.

Tip:

Double-click on an option's marker in the legend to change it to a different symbol or letter



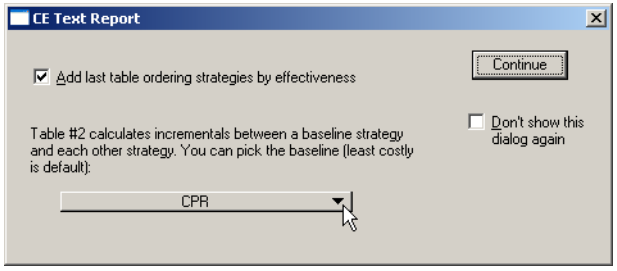
The CE analysis text report

To see the calculated values underlying the CE graph, open the graph's text report.

► **To display the cost-effectiveness analysis text report:**

▷ In the CE graph window, choose **Actions** > Text Report… (also in the **Graph** menu).

▷ Before displaying the text report window, TreeAge Pro prompts the user to select settings for the report (e.g., for extra table ordering strategies by increasing effectiveness).


CE Text Report
Add last table ordering strategies by effectiveness
Continue
Don't show this dialog again
Table #2 calculates incrementals between a baseline strategy and each other strategy. You can pick the baseline (least costly is default):
CPR

The text report for the Blindness Prevention decision is displayed below.

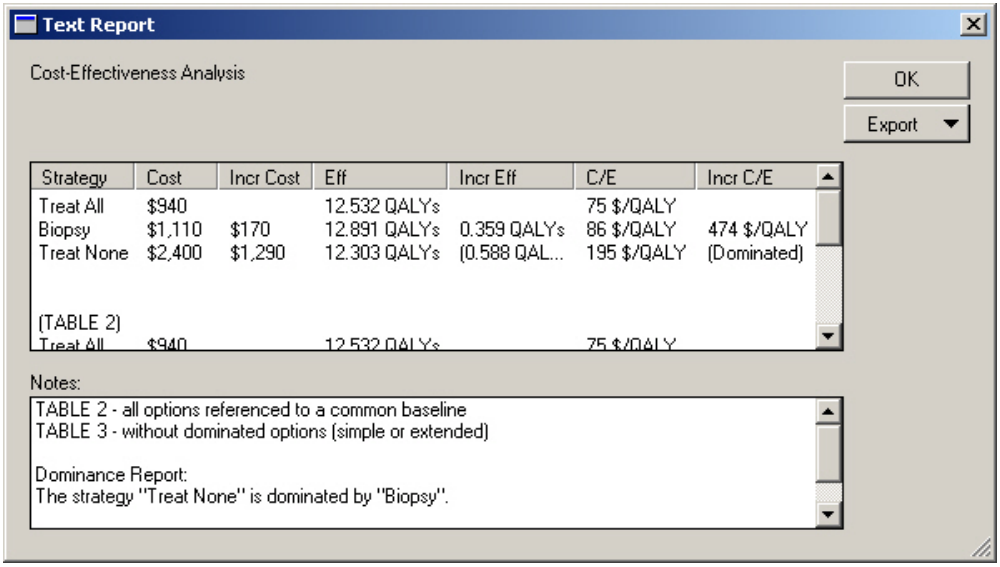
The first table in the report is the standard cost-effectiveness analysis table, showing the options in order of increasing cost, showing there average and incremental cost and effectiveness values, as well as incremental cost-effectiveness ratios.

Refer to the Notes section of the dialog for interpretation of additional tables in the report. For example, a second table shows incremental cost and effectiveness calculated between the baseline option specified in the text report setup dialog (default is still the *least costly* option) and every other option.

For more details on the interpretation of cost-effectiveness analysis results, continue reading this chapter, and refer to the cost-effectiveness analysis references listed at the beginning of the chapter.

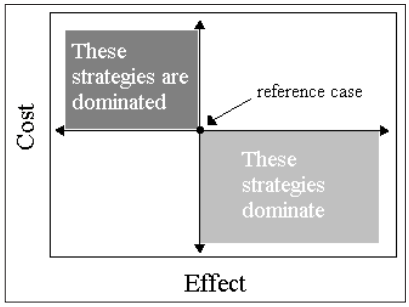
A text report, when generated from a graph window, uses the formatting of the graph axes. To change the formatting of the CE text report, modify the CE graph axes using the **Options** dialog.

The same text report can be displayed without generating the CE graph first, by instead running the **Analysis** > Rankings… command.


Text Report
Cost-Effectiveness Analysis
OK
Export
Strategy Cost Incr Cost Eff Incr Eff C/E Incr C/E
Treat All $940 12.532 QALYs 75 $/QALY
Biopsy $1,110 $170 12.891 QALYs 0.359 QALYs 86 $/QALY 474 $/QALY
Treat None $2,400 $1,290 12.303 QALYs (0.588 QAL... 195 $/QALY (Dominated)
(TABLE 2)
Treat All $940 12.532 QALYs 75 $/QALY
Notes:
TABLE 2 - all options referenced to a common baseline
TABLE 3 - without dominated options (simple or extended)
Dominance Report:
The strategy "Treat None" is dominated by "Biopsy".

Dominance and incremental cost-effectiveness

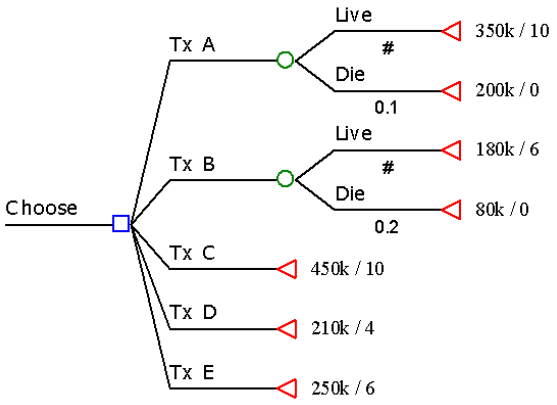
In a cost-effectiveness analysis, sometimes a strategy can be eliminated based on its relative cost and effectiveness compared to another strategy. An option is said to be *dominated* if it both costs more and is less effective than a comparator. This condition can be visually identified in a cost-effectiveness graph.

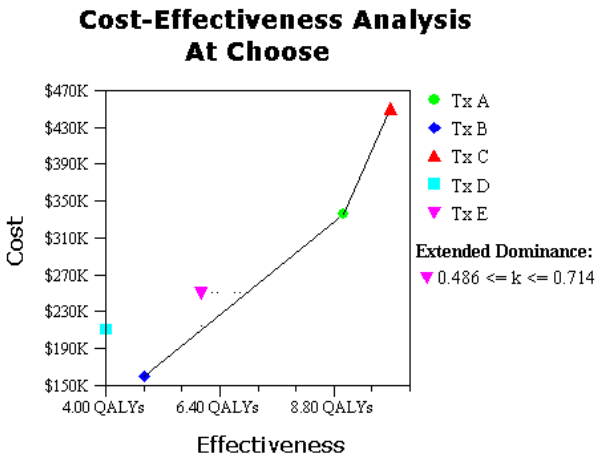


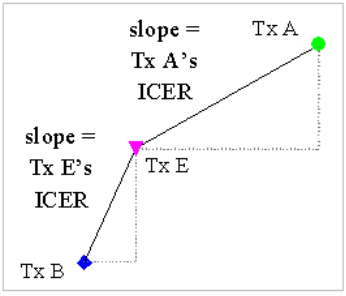
When effectiveness is plotted on the X axis, a strategy is absolutely dominated (sometimes referred to simply as *dominated*) if it lies above and to the left of another alternative. The option below and to the right is referred to as *dominant*, or *dominating*. Since an alternative that is dominated is often removed from the analysis, in TreeAge Pro such options are excluded from the cost-effective frontier.

In the CE graph shown on the previous pages, from the baseline analysis of the Blindness Prevention treatment decision, it is visually apparent that Treat None is dominated by Treat All. (However, we will see later that this may not be the case for all estimates of the uncertain or variable parameters, such as the prevalence of the disease, i.e., if a different population is modeled.)

Another, more complex CE graph is shown below. This graph, created from the Extended Dominance tree found in TreeAge Pro's Tutorial Examples\ Healthcare subdirectory, represents the cost and effectiveness of five strategies. The tree has been set up to illustrate special dominance conditions.







D is the only option removed from the cost-effective frontier based on absolute dominance — it is more costly and less effective than *B*, the least costly option. The cost-effective frontier is defined starting with *B*, skipping *E*, continuing to *A*, and then to *C* (the most costly and most effective option).

Why is strategy *E* excluded? It is *not* dominated in the absolute sense – visually, there is no option below and to the right. TreeAge Pro has, however, flagged *E* as being *extendedly dominated*, creating a special legend including *E* and also marking it in the graph area (with horizontal and vertical lines intersecting the cost-effective frontier). Interpreting special conditions of extended dominance in a CEA requires an understanding of the related concept of incremental cost-effectiveness ratios (ICERs).

Extended dominance and ICERs

A single cost-effectiveness analysis for a particular health condition takes place within a wider context, in which providing the best range of treatments and prevention is the goal, but limited financial, human, and other resources must eliminate some options.

In CEA, when comparing two, non-dominated options, an incremental cost-effectiveness ratio (ICER) is calculated. The ICER of the more effective option is the ratio of mean incremental cost and mean incremental effectiveness (e.g., in terms of $/QALY). Graphically, it is the slope of the line connecting two, cost-ordered strategies.

ICERs are used in the CEA process in a couple of ways. First, ICERs are used to determine whether options can be removed from the cost-effective frontier based on *extended* dominance. Lower ICERs correspond to better *value* (i.e., lower cost per unit of additional effectiveness). As shown in the partial graph at left, *A* is more effective than *E* and has a lower ICER (slope decreases). Thus, the cost-effective frontier connects *B* and *A*, but skips *E*, based on extended dominance.

With effectiveness on the horizontal axis, the slope of the line segment connecting two options corresponds to the ICER. Slopes approaching horizontal correspond to better (lower) ICERs. In the portion of the graph shown at left, it can be seen that Treatment A is more effective than Treatment E, and also has a lower ICER than Treatment E. In other words, it is a better *value* relative to Treatment B.

In addition to the visual indication of extended dominance provided in the CE graph, the text report provides details about conditions of extended dominance.

Strategy	Cost	Incr Cost	Eff	Incr Eff	C/E	Incr C/E (ICER)
Tx B	$160K		4.80 QALYs		33,333 $/QALY	
Tx D	$210K	$50K	4.00 QALYs	(0.80 QALYs)	52,500 $/QALY	(Dominated)
Tx E	$250K	$90K	6.00 QALYs	1.20 QALYs	41,667 $/QALY	75,000 $/QALY
Tx A	$335K	$85K	9.00 QALYs	3.00 QALYs	37,222 $/QALY	28,333 $/QALY
Tx C	$450K	$115K	10.00 QAL...	1.00 QALYs	45,000 $/QALY	115,000 $/QALY

Strategy	Cost	Incr Cost	Eff	Incr Eff	C/E	Incr C/E (ICER)
(TABLE 3)						
Tx B	$160K		4.80 QALYs		33,333 $/QALY	
Tx A	$335K	$175K	9.00 QALYs	4.20 QALYs	37,222 $/QALY	41,667 $/QALY
Tx C	$450K	$115K	10.00 QAL...	1.00 QALYs	45,000 $/QALY	115,000 $/QALY

Dominance Report:
The strategy "Tx D" is dominated by "Tx B".

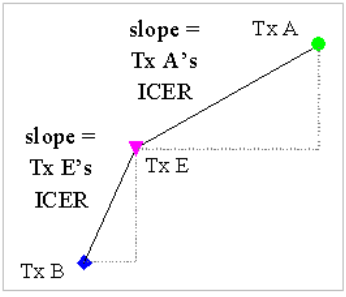
Extended Dominance Report:
The strategy "Tx E" is dominated by a blend of "Tx B" and "Tx A",
with a coefficient of inequity between 0.486 and 0.714.

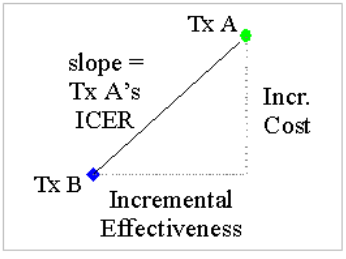
If a strategy has a higher ICER than the next more costly, more effective strategy in Table #1, it is extendedly dominated by that more cost-effective strategy. This is highlighted in Table #3, which excludes all strategies that are subject to extended or absolute dominance, and in the Notes section.

The threshold ICER (or willingness-to-pay, or ceiling ratio)

Note that Table #3 recalculates Treatment A's ICER after the extendedly dominated option, Treatment E, is removed. The new comparison is made with the baseline Treatment B, resulting in a new ICER for Treatment A which will be higher than the initial calculation in Table #1 (this is visually apparent in the graph on the previous page). This recalculation may be relevant in the second application of ICERs.

The second way ICERs are used is to determine if, at some point on the cost-effective frontier, the next more effective option exceeds a threshold ICER, sometimes referred to as the willingness-to-pay, or ceiling ratio. To efficiently allocate resources among competing priorities, there is normally a limit to the cost that a decision maker is willing to incur in order to gain an additional unit of effectiveness.





In the example, *A* initially has an ICER = 28,333 relative to *E*. After *E* is removed due to extended dominance, *A's* ICER is recalculated as 41,667, relative to *B*.

If the decision maker assumed a threshold ICER of 40,000, for example, then *A* would exceed this — it might be considered too costly for the additional effectiveness it offers.

Chapter 32 continues the discussion of threshold ICERs, in the context of sensitivity analysis in TreeAge Pro.

Extended dominance: an additional perspective

In some cases, strategy selection may involve not just maximizing effectiveness and working within a threshold ICER, but also working under a budget constraint (i.e., a cost threshold). If such a cost ceiling was set at $300K, in the example on the previous page, this would eliminate Treatment A (if it were not already eliminated based on an ICER threshold).

Theoretically, at least, if a decision maker is making a population-wide policy decision, two (or more) strategies might be combined to create a "blended" strategy that is less expensive (and less effective) than the too costly option. For example, instead of assigning Treatment A to all patients, they could be randomly assigned in some proportion to Treatment A and Treatment B. In the CE graph, this would result in a new strategy somewhere on the line connecting the two strategies.

The line connecting two options in the graph represents the average cost and effect for all possible blends of the two treatments. The intersection of the ICER line with the cost ceiling (a horizontal line) represents the best hypothetical option blending Treatment A and Treatment B. This optimal blend point is represented as *k*, calculated as the ratio:

(Cost A - Cost Ceiling)/(Cost A - Cost B)

The interpretation is that k% of patients treated are given the less effective Treatment B instead of Treatment A (or all patients are given Treatment B for k% of their treatments, and Treatment A the rest of the time). Questions of equity mean, however, that blends are not often relevant.

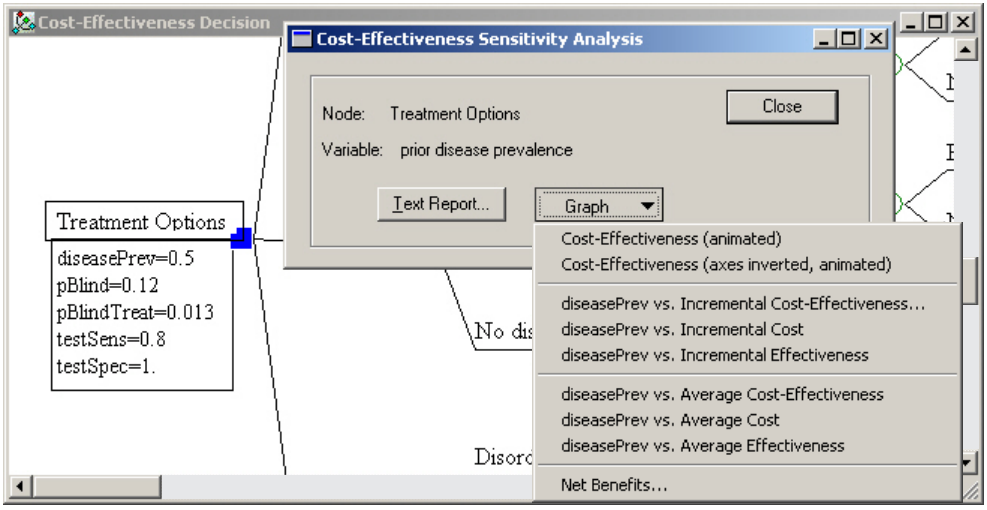
Extended Dominance:
▼ 0.486 <= k <= 0.714

The other aspect of the concept of blending, and the blend line, is related to the extended dominance concepts discussed in the previous section. If a blended strategy is created, it may cause an extendedly dominated strategy to become an absolutely dominanted one. In the extended dominance example on previous pages, some hypothetical blends of A and B would absolutely dominate Treatment E.

One-way cost-effectiveness sensitivity analysis

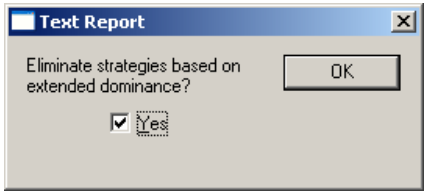
In a CE tree, the steps described in Chapter 12 can be used to perform a one-way sensitivity analysis.

Because of the added complexity of multiple attributes and incremental calculations, TreeAge Pro provides access to textual and graphical results of one-way, CE sensitivity analysis in an intermediate output window.



CE sensitivity analysis text report

The text report from a one-way CE sensitivity analysis shows, at every interval, the average and incremental values for each alternative. Essentially, this report repeats the information from Table #1 and Table #3 from the CE text report described above, for each interval of the sensitivity analysis.

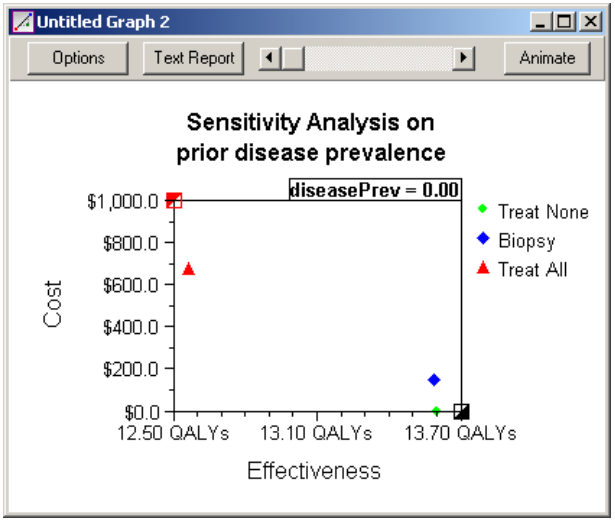


TreeAge Pro will prompt you to choose whether to calculate ICERs in the text report with extendedly dominated options left in place, or to mark them as extendedly dominated and exclude them from ICER calculations.

CE sensitivity analysis graphs

The Graph pop-up menu in the CE sensitivity analysis output window offers multiple ways to view the sensitivity analysis output graphically. The available graphs are described briefly below.

- **Cost-Effectiveness (animated)** – The results of the sensitivity analysis are presented as an animated version of the CE graph described above, with effectiveness on the horizontal axis. Each frame shows the CE graph for one iteration of the sensitivity analysis. Pressing the Animate button, or using the slider, causes TreeAge Pro to step through each interval.



- **Cost-Effectiveness (axes inverted, animated)**– Same as above, with the axes inverted; cost is on the horizontal axis.

There are seven other graph types available. Each resembles a simple one-way sensitivity analysis line graph, showing how the selected output value varies as a function of the input variable's changing value.

- ***Variable* vs. Incremental Cost-Effectiveness** – Before displaying this graph, TreeAge Pro prompts you to specify whether any dominated options' ICER values should be drawn as actual negative values (i.e., ratio of positive incremental cost to negative incremental effectiveness), or as zeros. The former is the default, since the baseline, least costly option this graph, is displayed as zero at an interval. See the section on finding ICER thresholds and 1-way CE sensitivity analysis graphs, in Chapter 32. Also review the Net Benefits graph option, below.
- ***Variable* vs. Incremental Cost**
- ***Variable* vs. Incremental Effectiveness**
- ***Variable* vs. Average Cost-Effectiveness**
- ***Variable* vs. Average Cost**
- ***Variable* vs. Average Effectiveness**
- **Net Benefits… –** This graph is very useful for identifying cost-effective thresholds, particularly in models with small differences in effect, changing cost ordering, and/or more than two strategies. Chapter 32 provides details on using the net health benefits (NHB) and net monetary benefits (NMB) graphs in the TreeAge Pro Healthcare module.

Chapter 32

Cost-Effectiveness Modeling and Analysis Options

The previous chapter provided instructions on preparing a tree for cost-effectiveness calculations, performing baseline cost-effectiveness analysis, and running cost-effectiveness sensitivity analysis.

This chapter covers a variety of other useful cost-effectiveness modeling preferences and techniques available in the TreeAge Pro Healthcare module, as well as additional cost-effectiveness analysis features.

In this chapter:

- Net benefits calculations 408
- Multi-attribute weighted costs 409
- Inverting effectiveness calculations 413
- CE roll back optimal path parameters 416
- Thresholds and CE sensitivity analysis 418
- Displaying incremental values during roll back 425

Net benefits calculations

The calculation of net monetary benefits (NMB) and net health benefits (NHB) is increasingly prevalent in health economic evaluations, either in addition to or sometimes instead of the usage of ICERs.

The net monetary benefit (NMB) of an alternative is calculated using the following formula:

$$NMB = E * WTP - C$$

where E represents effectiveness, C represents cost, and WTP is the willingness to pay (i.e., the decision maker's threshold ICER).

The net health benefit (NHB) of an alternative is calculated using a similar formula:

$$NHB = E - C / WTP$$

Some advantages of using net benefits:

- Regardless of the number of strategies, the most cost-effective comparator is simply the one with the highest net benefit, given a threshold ICER ("WTP" in TreeAge Pro).
- When trying to describe uncertainty in CE models with small mean differences in effectiveness (or many competing alternatives), net benefit calculations are not unstable as ratios like the ICER can be.
- Net benefits are the basis for acceptability curves and value of information curves.

Additional background on the net benefits framework and analyses can be found in various journal articles, including:

"Quantifying stochastic uncertainty" Glick, Briggs, and Polsky, *Expert Rev Pharmacoeconomic Out Res 1(1), 25-36 (2001)* [and www.future-drugs.com].

"Net Health Benefits," Stinnett and Mullahy, *Med Decis Making* 18 (1998) supplement: S68–S80.

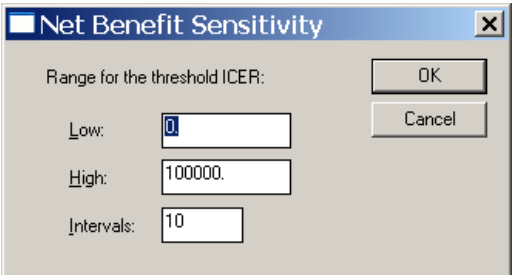
Performing net benefits calculations

In the TreeAge Pro Healthcare module, graphs and reports based on NHB or NMB calculations are available in sensitivity analysis, tornado diagrams, and Monte Carlo simulation of cost-effectiveness models; more details are provided later in this chapter and in the next chapter.

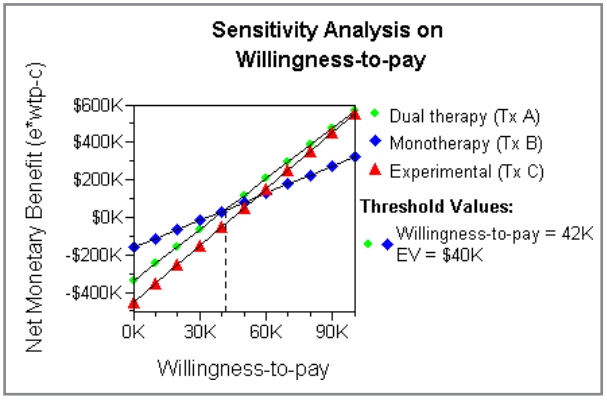
TreeAge Pro 2007 adds a baseline Net Benefits analysis, as well.

► **To do a basic Net Benefit curve:**

▷ In a Cost-Effectiveness tree, select a decision node (or a single strategy) and choose **Analysis** > Net Benefits….



▷ Specify a range of willingness-to-pay (in other words, weight on effectiveness). Using a Low of 0 will reflect putting no weight on effectiveness (i.e., requiring cost savings). Using an extreme value for High will, at the other end of the graph, reflect decision making when putting very little weight on cost.

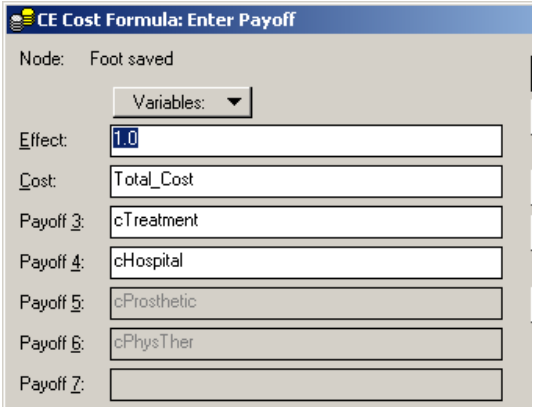


Multi-attribute weighted costs

TreeAge Pro provides access to 9 payoff/reward sets. In Multi-Attribute calculations (see Chapter 7 for details), this means that up to 9 payoffs can be combined using a weighting function — i.e., a set of numeric or variable ***weights*** corresponding to payoff or Markov reward sets.

Under the Cost-Effectiveness calculation method, TreeAge Pro allows you to use a weighted cost function in the same way. Up to 8 payoff sets can be combined as the net cost component of CE calculations.

Using a weighted cost function in a cost-effectiveness model may facilitate clearer identification of the parts of a complex cost formula (for example, drug costs, hospital costs, and inpatient costs). It also makes it easier to switch between CEA using a single cost component, and CEA using different combinations of cost components.



Multi-attribute costs: an example

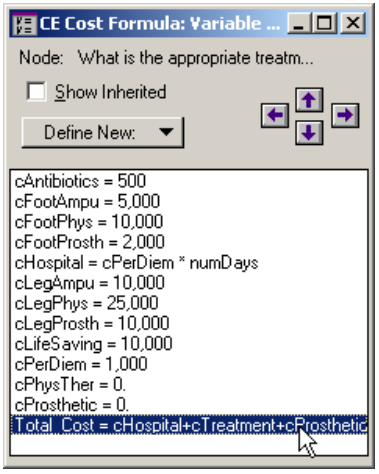
In the example trees in the previous chapter, simple numeric values were assigned to each terminal node's cost payoff. In most models, however, cost payoffs or rewards will be more complicated. The tutorial in Chapter 14 includes an example tree in which a more realistically complex cost formula is used. The Cost Formula tree uses variables to represent the components of a cost formula. In the tutorial, the Simple calculation method was used, but the same issues apply to cost calculations under the Cost-Effectiveness calculation method.

Instead of representing the costs of hospitalization, surgery/drugs, prosthetics, and physical therapy as components of a single payoff (#1 in the example), each of these components can instead be placed in a separate payoff (i.e., #3-#6). Under the Multi-Attribute calculation method or the Cost-Effectiveness calculation method with multi-attribute costs, a simple weighting function could be used to recombine the component variables into a single cost calculation.

To see how the multi-attribute cost weightings work, open the example tree called CE Cost Formula, found in TreeAge Pro's Tutorial Examples\ Healthcare subdirectory. This is a CE version of the Cost Formula example tree.

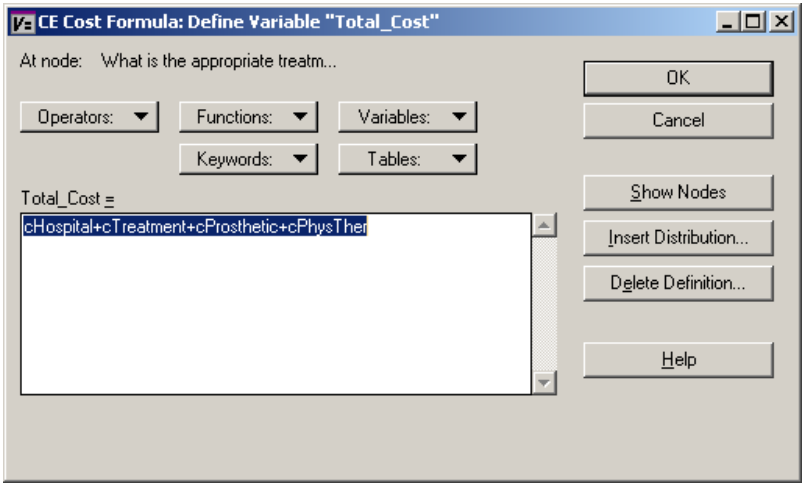
If you look at the payoffs for a terminal node, you will see that the current calculation preferences are using the Total_Cost variable for cost calculations. It is also possible to see that four other payoffs (#3 through #6) have been assigned the individual components of the Total_Cost formula.

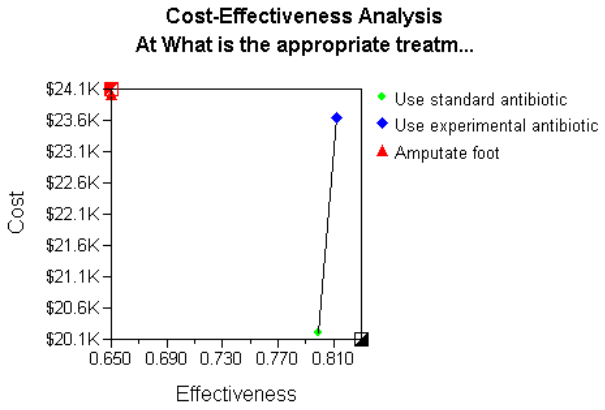
To inspect the cost formulas, open the Variables Definitions window. Select the root node and choose **Values >** Show Variable Definitions Window.



You can see that

$$\text{Total\_Cost} = \text{cTreatment}+\text{cHospital}+\text{cProsthetic}+\text{cPhysTher}.$$



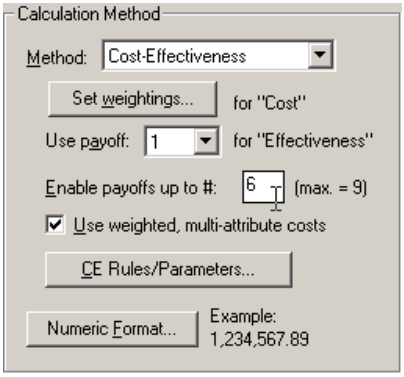


Before making any changes to the tree, perform a cost-effectiveness analysis at the decision node. Later, after changing the payoff calculation preferences in the tree, we can re-run the analysis and compare the results to ensure that no errors were made.

Now, modify the Calculation Method preferences to use a multi-attribute cost formula, combining the four cost components already in payoffs #3 through #6.

► **To set up a weighted multi-attribute cost function:**

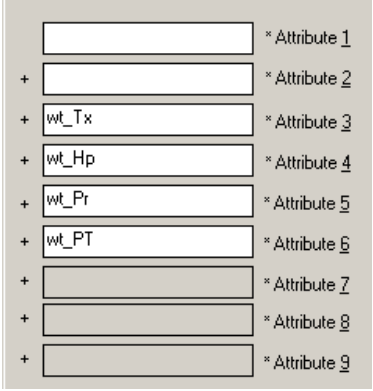
▷ Choose **Edit >** Preferences…. In the Calculation Method category, select Cost-Effectiveness calculations and check the option labeled Use weighted, multi-attribute costs.



▷ Change the Enable payoffs… setting to a sufficient number of payoffs for the multi-attribute cost formula; in this case enter `6` to enable payoffs 5 and 6.

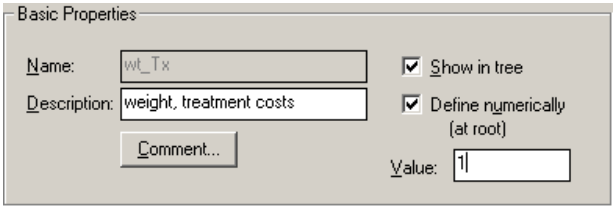
▷ Click the **Weightings…** button.

▷ Either enter weights of `1` for each payoff that should be added together for the cost calculation, or enter variables for greater flexibility, as shown at left. Note that the box corresponding to the effectiveness payoff should be left blank (along with any unneeded payoffs).



▷ Click OK to store the weights, and return to the Preferences dialog.

If you utilize any variables in the weighting function, they should normally be defined once, default for the tree.



The variables will be dynamically reevaluated at each calculated terminal node or Markov node. Sensitivity analysis can be performed on variables used for weights.

▷ Click OK to close the Preferences dialog and save the Cost-Effectiveness preferences changes.

Before trying any other analyses on this example tree, re-run the Cost-Effectivenesss analysis at the decision node to verify that the preferences changes work as expected, and that the CE graph looks the same as before turning on the multi-attribute cost calculation settings.

Notes on using multi-attribute costs in CE trees

When weighted, multi-attribute costs are in use in a cost-effectiveness model, terminal nodes will display the weighted cost payoff formula in parentheses; nodes in a Markov model will display separate Markov information line items for each cost reward set.

To turn off the display of multi-attribute payoff expressions, open the Terminal Nodes/Columns preferences category, and uncheck the option labeled Display payoff names.

If you do not enter terminal node payoff expressions for each active cost attribute (i.e., each payoff that is enabled and has an assigned multi-attribute weight), calculation errors will occur. If you enter a weight for the payoff set currently assigned to effectiveness, it is simply ignored during cost calculations. If you leave a weight blank, it evaluates to 0. If you subsequently reduce the number of enabled payoffs in the Calculation Method preferences, any disabled payoffs will be excluded from the weighted cost calculation.

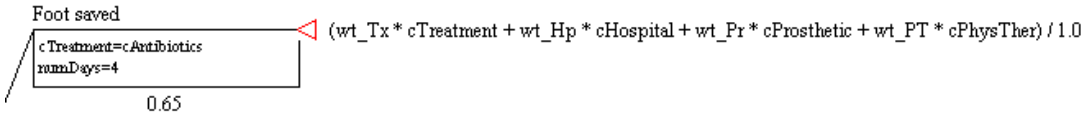
The same weighting function will apply if the calculation method is changed to Multi-Attribute, instead of Cost-Effectiveness.

Markov CE models using multi-attribute costs

Just as with regular trees, Markov models can use the Multi-Attribute calculation method or the Cost-Effectiveness calculation method with the multi-attribute cost preferences described above.

Previously in the DATA software, multi-attribute Markov models were handled differently in expected value calculations and simulation trials. TreeAge Pro however calculates each Multi-Attribute (or Cost-Effectiveness with multi-attribute costs) Markov process in a single pass, with the cost weighting done during the Markov process, and using either the reward set #1 termination condition (for Multi-Attribute) or the CE termination condition. See, for example, the Multi Cost Markov tree in the Tutorial Examples\Healthcare subdirectory.

See the chapters on Markov modeling later in this manual for more details.


Foot saved
cTreatment=cAntibiotics
runnDays=4
0.65
(wt_Tx * cTreatment + wt_Hp * cHospital + wt_Pr * cProsthetic + wt_PT * cPhysTher) / 1.0

Inverting effectiveness and incremental effectiveness calculations

A basic assumption in most CE models is that when it comes to effectiveness, higher numbers are always better. When a cost-effectiveness model presents the reverse situation, with lower values of the effectiveness attribute being preferable, you must invert the calculated effectiveness and/or incremental effectiveness calculations in the tree.

Consider a cost-effectiveness study which tracks the number of adverse events as its measure of effectiveness. In this case, the alternative with the *lowest* calculated effectiveness value is the most effective treatment. By default, however, TreeAge Pro's CE analysis algorithm normally identifies options with *higher* calculated effectiveness value as being preferred. In the case of the adverse event model, leaving this default behavior will result in a CE graph and text report which incorrectly calculates incremental effectiveness and determines dominance.

Net Benefits:

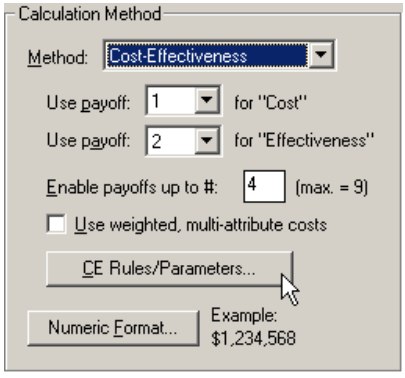
To calculate Net Benefits in trees using inverted incremental effectiveness, TreeAge Pro simply reverses the sign on effects in the *E*WTP - C* formula. This means that Net Benefits are always negative, but the strategy closest to 0 is optimal.

Inverting incremental effectiveness

The simplest, and usually clearest, option for ensuring that CE calculations correctly interpret effectiveness measures that should be minimized is to invert only the *incremental* effectiveness calculations performed at decision nodes, leaving the reporting of effectiveness values unchanged. The example tree used here is called Invert CE Markov.

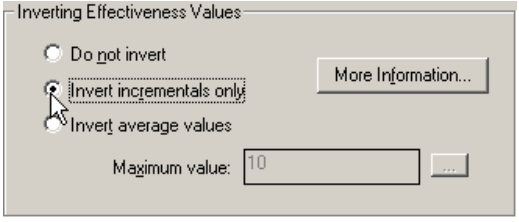
► **To invert incremental effectiveness calculations only:**

▷ Choose **Edit >** Preferences, and go to the Calculation Method category in the Preferences dialog.



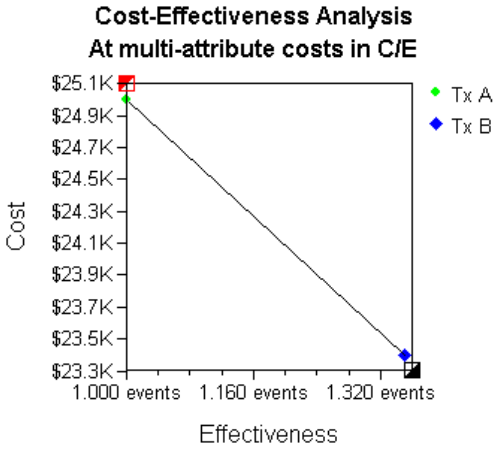
▷ Confirm that the Cost-Effectiveness calculation method is selected, and click the **CE Rules/Parameters...** button.

▷ In the Cost-Effectiveness Parameters dialog, click the option labeled Invert incrementals only.



▷ Click OK to save the CE parameters, and click OK again to exit the Preferences dialog and update the tree.

Inverting incremental effectiveness calculations results in a correct construction of the cost-effective frontier (with lines sloping down).



Analyses that use or report incremental effectiveness values will simply reverse the normal assumptions, instead calculating how much less "effective" each more preferable option is.

Strategy	Cost	Incr Cost	Eff	Incr Eff	C/E	Incr C/E (ICER)
Tx B	$23.4K		1.350 events		17,333 $/event	
Tx A	$25.0K	$1.6K	1.000 events	0.350 events	25,000 $/event	4,571 $/event

Inverting incremental effectiveness is usually the preferable method of dealing with an inverted measure of effectiveness, as it does not complicate the reporting of expected effectiveness, and does not require specifying a maximum effectiveness value, as does the following, alternate method.

Inverting expected effectiveness calculations

The second method works by inverting the expected effectiveness values calculated for each node, thereby resulting in inverted incremental effectiveness values, as well

► **To invert all nodes' effectiveness values:**

▷ In the Cost-Effectiveness Parameters dialog, click the option labeled Invert average values.

▷ Assign a fixed maximum effectiveness value from which to subtract all nodes' calculated effectiveness values during cost-effectiveness calculations.

▷ Click OK or to save the CE parameters, and again to exit the Preferences dialog.

The inversion of calculated effectiveness values (rather than just incremental values, as above) results, again, in a correct ordering of options in the CE report, shown below. The options' calculated incremental effectiveness values (and incremental CE ratios) are the same for both methods.

When inverting all nodes' calculated effectiveness values, a maximum effectiveness value should be selected which is greater than or equal to the uninverted effectiveness of any particular option (even during sensitivity analysis). This will ensure that no inverted values are negative. You may assign a variable or expression for the maximum; this expression will be evaluated prior to analysis, at the root node.

Effectiveness is not inverted *within* a Markov process. TreeAge Pro will invert the Markov node's calculated effectiveness, but Markov analysis graphs and text reports will use *uninverted* effectiveness values.

CE roll back optimal path parameters

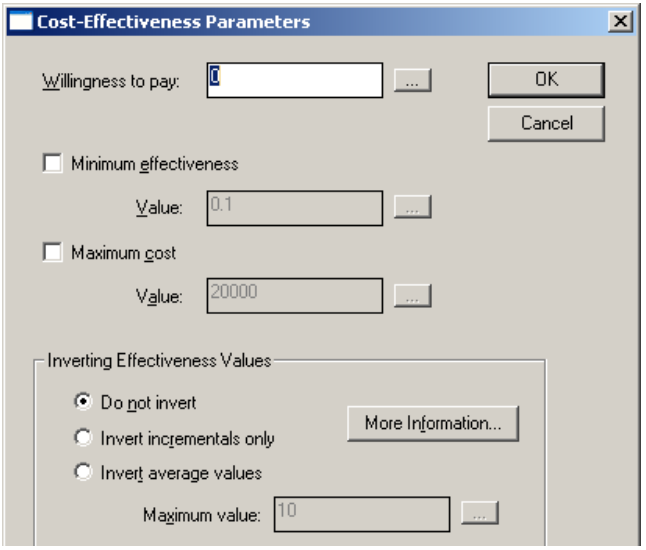
As described in the previous chapter, the standard method of performing a baseline CEA in the TreeAge Pro Healthcare module is using the **Analysis >** Cost-Effectiveness… or > Rankings… commands at a decision node. In some cases, however, it may be useful to also display calculated CEA values in the rolled back tree.

In order to roll back a CE tree, TreeAge Pro must be able to automatically select an optimal path at decision nodes. During CE roll back (and any analysis of a CE tree with embedded decision nodes) TreeAge Pro uses a number of special preferences that enable the model builder to customize the roll back algorithm.

► **To set the optimal path parameters for cost-effectiveness calculations:**

▷ Choose **Edit >** Preferences to open the Preferences dialog, and display the Calculation Method category.

▷ Press the **CE Rules/Parameters…** button.



While it is possible to set the CE parameters so that TreeAge Pro makes decisions simply by minimizing the CE ratio, it also possible to have TreeAge Pro do the following:

- eliminate options below a minimum effectiveness;
- eliminate options above a certain cost; and
- select the most effective option within an incremental cost-effectiveness threshold.

It is also possible to invert the effectiveness measure, as described earlier in this chapter.

▷ In the Cost-Effectiveness Parameters dialog, enter the information used to evaluate an optimal path at a decision node during roll back.

▷ Press OK to accept the entered parameters.

Tip:

Clicking on the ellipses buttons next to the parameter entry boxes will open an expression editor dialog, where formulas using variables and functions can readily be set up. Expressions entered for the cost-effectiveness parameters will always be calculated at the root node of the tree, regardless of the location of the decision node being evaluated.

The CE optimal path algorithm

In essence, the optimal alternative will be the most effective option with an ICER not greater than the specified willingness-to-pay (WTP). Any analysis that must select an optimal path from among the strategies at a decision node in a CE tree, does so using the following algorithm.

First, if minimum effectiveness or maximum cost constraints are specified, any option that fails either test is eliminated. Second, the remaining options are ordered by increasing cost. Third, any option which is dominated by another less costly, more effective option is eliminated. Fourth, each option whose ICER calculated relative to the next least costly option is greater than your WTP criterion is eliminated. Finally, the most effective remaining alternative is selected as optimal.

If all options fail these tests, then the least costly option will be selected as optimal.

> ***Tip:***
> If a WTP of 0 is specified, as is the default, ICERs are ignored, and the least costly option will be selected. If a negative WTP is specified, TreeAge again ignores ICERs, but instead picks the option with the lowest C/E ratio.

Thresholds and CE sensitivity analysis

Chapter 12 covers the use of variables and sensitivity analysis in decision trees. The previous chapter in this manual briefly described the output of 1-way sensitivity analysis in CE decision trees. This section describes in more detail one aspect of CE sensitivity analysis — finding thresholds.

In a tree, threshold analysis involves searching an uncertain variable's range for values where there is a change in optimal strategy. In a CE model, this means identifying variable values where an alternative changes from being cost-effective to being non-cost-effective (or where the most cost-effective strategy changes).

For a single uncertain variable, CE thresholds can be identified in a number of ways:

- Run a 1-way sensitivity analysis and using either a Net Benefits (NHB or NMB) graph or the Incremental CE graph.
- Run a tornado diagram. Net Benefits calculations are used by default in the series of one-way sensitivity analyses.

Starting with v2007, multi-way CE sensitivity analysis can be used to find thresholds when varying 2 or 3 variables simultaneously. By default, Net Benefits are used in dividing graph regions and thus indicating thresholds between regions. As in previous versions of TreeAge Pro, a 2-way analysis on a decision with 2 alternatives can instead be performed with special ICER isocontour, or threshold, lines (and no Net Benefits calculations).

Net Benefits thresholds in a 1-way analysis

Given a particular WTP, a Net Benefits graph is a simple-to-use CE threshold analysis tool. As described at the beginning of the chapter, given the same WTP, the intervention with the highest net benefit (monetary or health) is the most cost-effective.

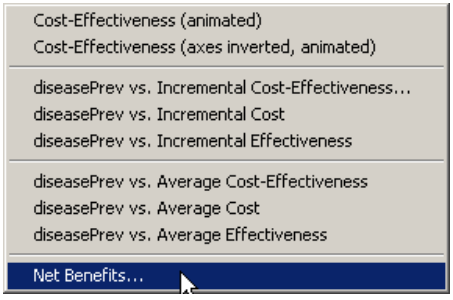
> ***Note:***
>
> As described at the beginning of the chapter, the calculations used for net monetary and net health benefits are:
>
> NMB = E * WTP - C
>
> NHB = E - C / WTP

TreeAge Pro lets you select which scale to use for the vertical axis in the Net Benefits graph, NMB or NHB.

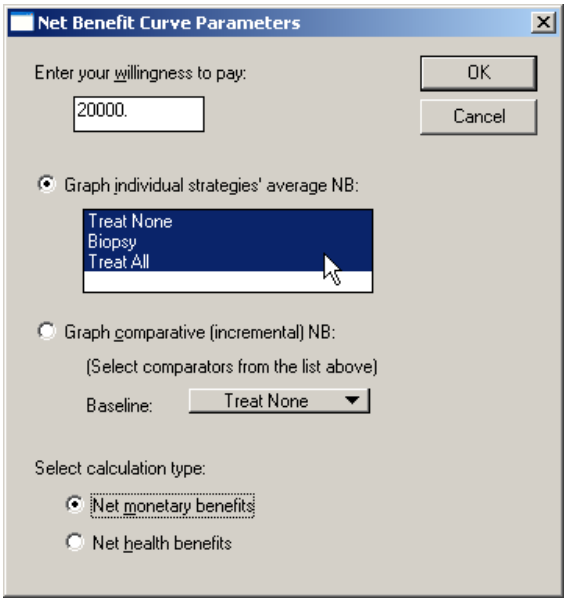
► **To identify thresholds using the Net Benefits graph:**

▷ Perform a one-way CE sensitivity analysis. (The example shown on the next page uses the CE version of the Blindness Prevention tree, analyzing the variable prev.)

▷ Once the analysis completes Select the Net Benefits… graph from the list.

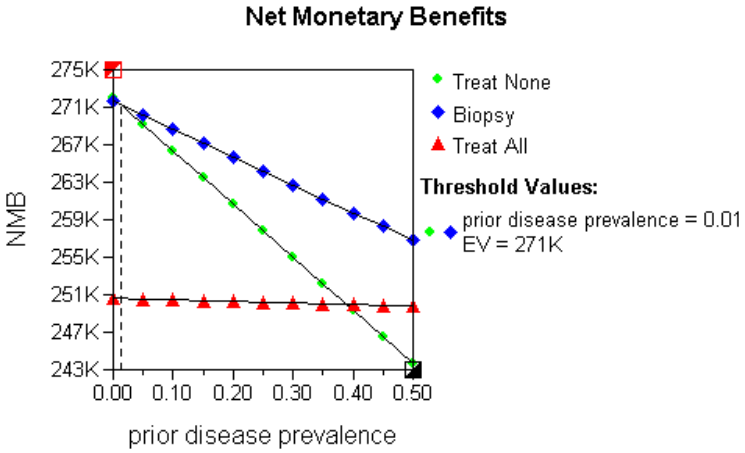


- Specify a value for willingness-to-pay (i.e., a ceiling ICER value), select all strategies for inclusion, and choose either Net Monetary or Net Health Benefits for the vertical axis scale.



A line graph will display the NMB or NHB function for each strategy.

If one option's line is always highest on the vertical, benefits scale for a particular analysis, there are no CE thresholds in that case. In this example, however, based on a WTP of 20000, the lines for Treat none and Biopsy exchange places as the option with the highest net benefit near prev=0.01. (For higher precision thresholds, increase the axes' decimal places in the Graph Options dialog. TreeAge Pro updates the Threshold Legend and the text report based on the axes' numeric formatting.)



The Net Benefits line graph's threshold analysis works for comparisons of any number of strategies.

Creating an incremental cost-effectiveness tornado diagram

Starting with TreeAge Pro 2005 release 1.x, it is possible to create a tornado diagram that reports the changing incremental value between two selected strategies. In a cost-effectiveness sensitivity analysis, this can be used to report incremental cost-effectiveness (rather than the preferred, default Net Benefits reporting, as described in the next section).

► **To create an incremental tornado diagram:**

▷ Instead of selecting a decision node, select two of its branches. To do this, click on one branch and then shift-click on a second branch.

▷ Then choose **Analysis >** Sensitivity > Tornado Diagram.

Using a Net Benefits tornado diagram

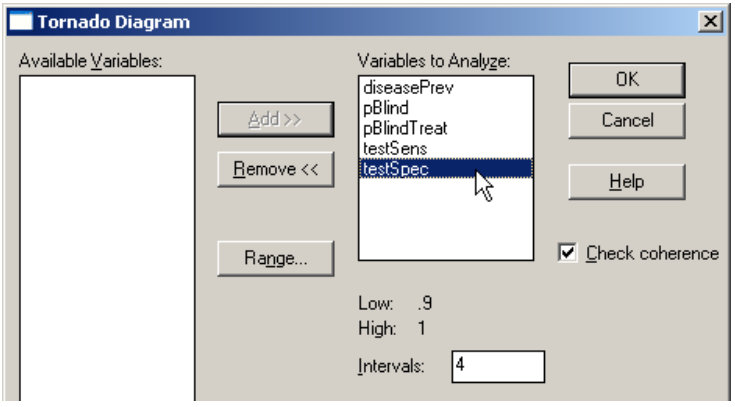
In a tornado diagram on a CE tree, TreeAge Pro normally calculates net benefits (see the beginning of this chapter for details). You can choose which net benefits scale to use for the horizontal axis, monetary or health.

Chapter 15 describes the use of tornado diagrams in detail. This section describes their particular application in CE trees.

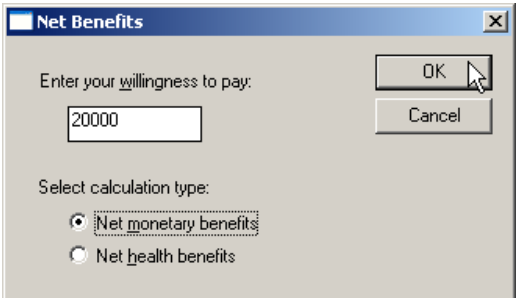
► **To identify thresholds in the Net Benefits tornado diagram:**

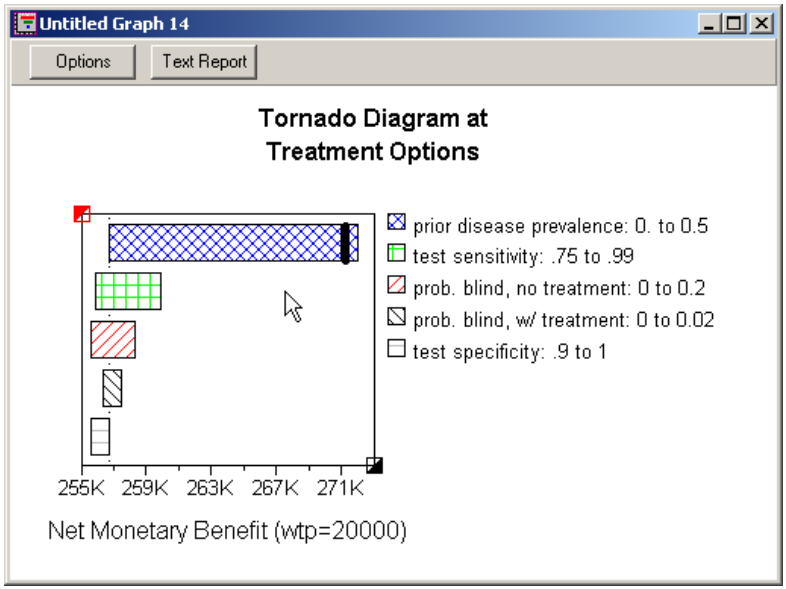
▷ Select the desired node for analysis, and choose **Analysis >** Tornado Diagram…. (The example shown again uses the CE version of the Blindness Prevention tree.)

▷ Add the desired variables for analysis, specifying their ranges and click OK.

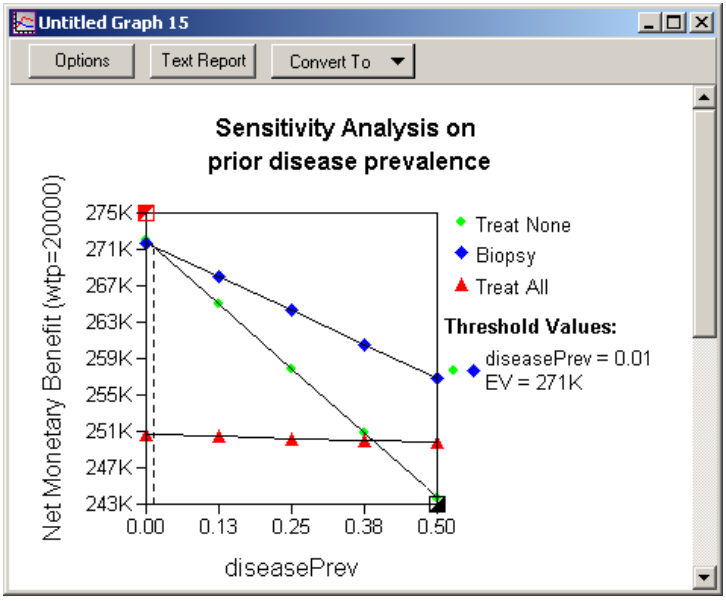


▷ Enter a threshold ICER (willingness-to-pay), and select Net monetary benefits or Net health benefits for the y-axis scale.





Once the analysis is complete, tornado bars will display for each variable, showing how the net benefit of the optimal alternative changes, as well as identifying threshold points with a heavy vertical line. Each bar is rooted on a vertical dotted line indicating the net benefit calculated for the baseline optimal alternative. Double clicking on a bar will display the underlying one-way sensitivity analysis graph.

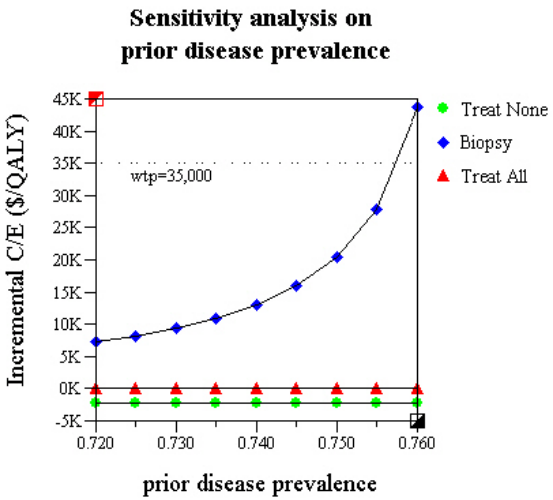


Finding cost-effective thresholds using the ICER graph

Using TreeAge Pro's incremental CE sensitivity analysis line graph, it is usually possible to find CE thresholds in a more complex way than that covered in the previous section on net benefits.

The Incremental CE graph shown below is from a one-way sensitivity analysis on the CE version of the Blindness Prevention tree used in the previous chapter. The graph displays the ICERs of three alternatives. For the moment, ignore the two alternatives with horizontal (zero and negative) lines; see below on interpreting zero and negative ICER values in the graph.

The rising curve for Biopsy represents its changing ICER, calculated at each interval relative to the next least costly non-dominated alternative. A dotted horizontal line has been added to the graph (see below for instructions) to help visualize a WTP (ceiling ratio) of 35,000 $/QALY. TreeAge Pro will also use the line to approximate the threshold variable value in the Notes section of the text report.



The intersection of the dotted WTP line and the curve representing Biopsy approximates a CE threshold — the prevalence value at which Biopsy changes from being cost-effective to being non-cost-effective, based on a threshold ICER of 35,000, is around 0.76.

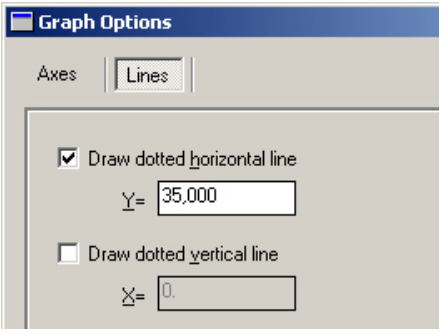
If you vary the WTP (by changing the y-axis value used to draw the dotted line), you can see how the threshold value of the sensitive variable changes.

► **To view the ICER threshold information:**

▷ Perform a 1-way CE sensitivity analysis, for instance on the prevalence variable in the Blindness Prevention tree.

▷ In the analysis output window, click on the Graph menu button and choose *variable* vs. Incremental Cost-Effectiveness (ICER).

▷ In the graph, click on the **Options** button (or choose Graph > Options) and then click on the Line tab in the Options dialog. Select the option labeled Draw dotted horizontal line, and specify a Y value equal to the desired willingness-to-pay threshold. Click OK to return to the graph.



▷ Optionally, use the **Graph >** New Label… command to add a custom graph label with text describing the threshold line (see Chapter 6 for more on custom graph labels).

▷ Click the graph's **Text Report** button. The Notes section of the text report will describe any point at which an option's ICER line crosses the dotted WTP line.

Notes:
Lines crossing your indicated threshold (35K):
Biopsy crosses at 0.75724

The following duplicate output may offer better import into other softwa

prior disease prevalence	0.72	0.725	0.7

In some cases, false crossing points may be reported because of a change in sign of the ICER's denominator (incremental effectiveness); see below for details.

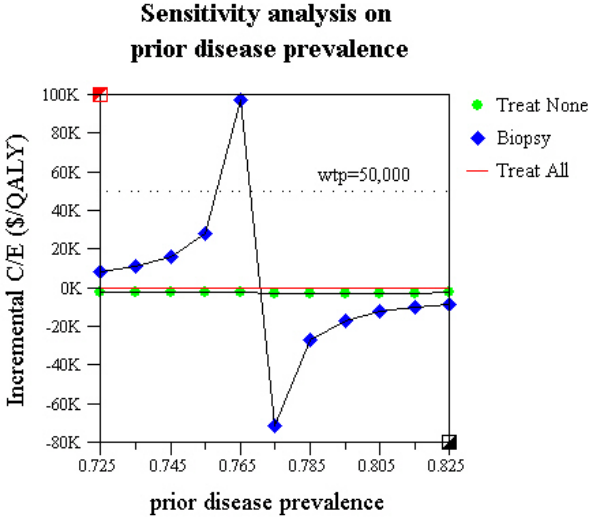
Caveats on using ICER threshold approximation

Threshold values included in the text report are found by linear interpolation (as in non-CE sensitivity analysis line graphs). The accuracy of the threshold approximation in the ICER graph can be improved, if the cost-effective option's line is curved, by using more intervals in the initial analysis.

In some cases, however, no number of intervals will resolve problems with the incremental CE ratios represented in the graph. As noted above, if the incremental effectiveness denominator of an alternative's ICER changes sign in an interval (positive to negative), crossing points of the horizontal WTP line in that interval do not represent true CE thresholds as described above. Instead, they indicate an effectiveness threshold — a point where the two strategies have equal effectiveness — and a change from an alternative being simply non-cost-effective to being absolutely dominated. This is illustrated in the graph below.

In using line segments to draw each option's curve in a sensitivity analysis line graph, TreeAge Pro assumes that each is a continuous function; TreeAge Pro simply connects each point in sequence. In cases where an alternative's incremental effectiveness changes from positive to negative, an alternative's ICER function is *asymptotic* – that is, it approaches positive infinity and negative infinity from opposite directions, as in the graph shown above. In the graph shown above, of the two crossing points indicated, only one (at prev=0.75818) is a valid threshold.

While the full text report of the CE sensitivity analysis shows blank cells for dominated alternatives rather than displaying the negative ICER values, the ICER graph (and its text report) can use the negative values. At every analysis interval of the ICER graph, the least costly (baseline) option will be represented with a zero value. If you choose to display dominated options as zero values, as well, this makes it difficult to visually distinguish the dominated options. Thus, the option of using the negative CE ratios for dominated options (i.e., options having a negative incremental effectiveness at some point) is useful when you have dominated options in the analysis.

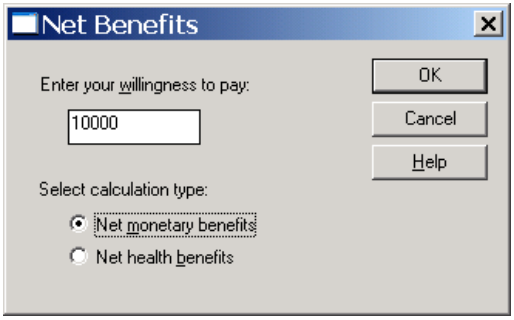


2- and 3-way CE sensitivity analysis thresholds using Net Benefits

As described in Chapter 15, in a tree set to calculate a single attribute (e.g., cost or utility), a two-way sensitivity analysis identifies the optimal alternative for each combination of values of the two variables. Based on this, a region graph is created in the two-dimensional variable space with regions assigned to the alternatives based on their optimality. The lines dividing two regions are threshold lines.

Starting with TreeAge Pro 2007, in cost-effectiveness trees it is possible to run 2- and 3-way sensitivity analysis using Net Benefits calculations, making CE thresholds easy to identify. Unlike previous versions, isocontours are not required and the graphs can be interpreted for any number of alternatives (not limited to two).

TreeAge will prompt you for a willingness-to-pay value to use in the Net Benefits calculations for all strategies (e.g., NHB = E*WTP - C).



For the given WTP, the strategy having the highest net benefit for any coordinate in the analysis will be assigned that point. Regions of cost-effectiveness can then be constructed on this basis.

(Note: Custom isocontours added to the graph will represent incremental net benefits, not incremental CE ratios.)

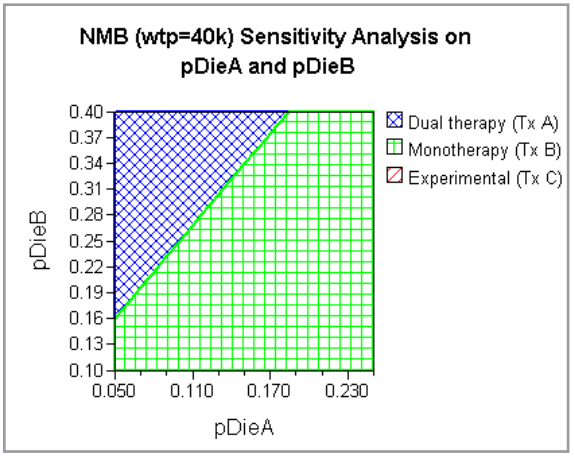
2-way CE sensitivity analysis (cost regions and isocontours)

If you press Cancel at the Net Benefits prompt when running a 2-way CE sensitivity analysis, TreeAge Pro will revert to a cost-only presentation of the region graph (as in v2006 and earlier).

In this form of the graph, regions are allocated simply based on lowest cost. If two alternatives are assigned partial regions, the line dividing them represents *zero incremental cost* (equal cost), and thus an ICER of zero.

If the decision has just two alternatives, custom isocontours (see Chapter 15) can be added via the Options dialog to identify CE thresholds. An isocontour represents a constant incremental value (ICER in this case) of the *topmost branch* of the decision node.

See the discussion of two-way sensitivity analysis region graphs in Chapter 15 for more on isocontours.



Displaying incremental values during roll back

The calculated incremental values of competing strategies in a cost-effectiveness tree are *not* automatically displayed when the tree is rolled back. The easiest way to report incremental expected values is to generate the CE analysis text report, as described in the previous chapter. It is possible, however, to display incremental values in the rolled-back tree.

For a visual display of incremental values in the rolled-back tree, you need to create terminal node columns that display the appropriate incremental values, and then collapse the subtrees to the right of the decision node. Terminal node columns, covered in detail in Chapter 11 can be used to display incremental values to the right of *visual* end nodes (not just terminal nodes) during roll back.

CHAPTER 33

COST-EFFECTIVENESS SIMULATION REPORTS AND GRAPHS

Chapter 16 covers the basic aspects of performing Monte Carlo simulation, and using the simulation output window to display statistical information, a full text report, and graphs describing the probability distribution of inputs and outputs.

The Healthcare module adds a number of graphs and reports designed specifically for Monte Carlo simulation of cost-effectiveness models. These are described in this chapter.

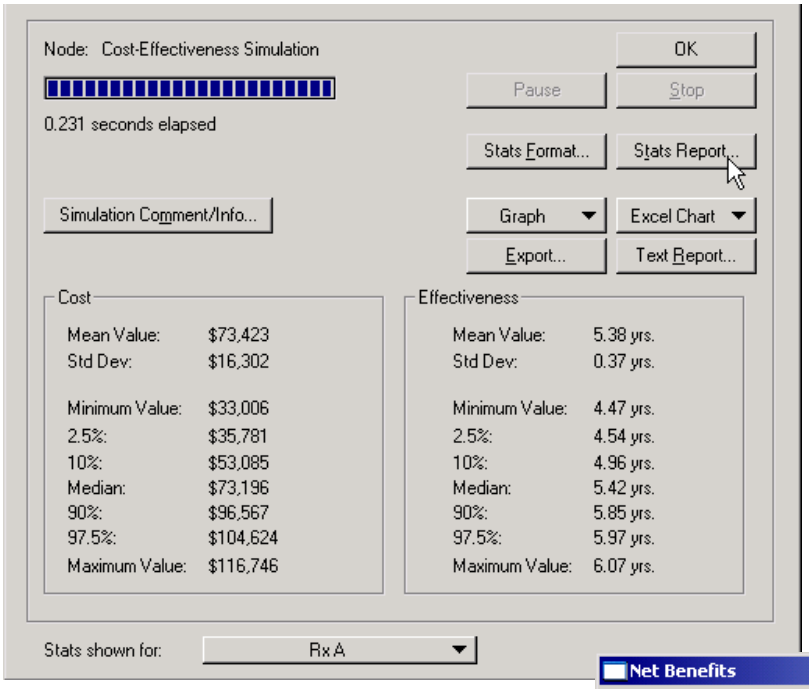
In this chapter:

- Basic CE statistics and simulation outputs 428
- Scatterplots..... 431
- Acceptability curves and Net Benefits calculations 436
- Cost-effectiveness value of information (EVPI and EVPPI)..... 441

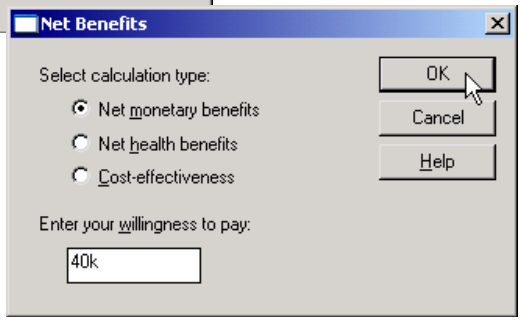
Basic CE statistics and simulation outputs

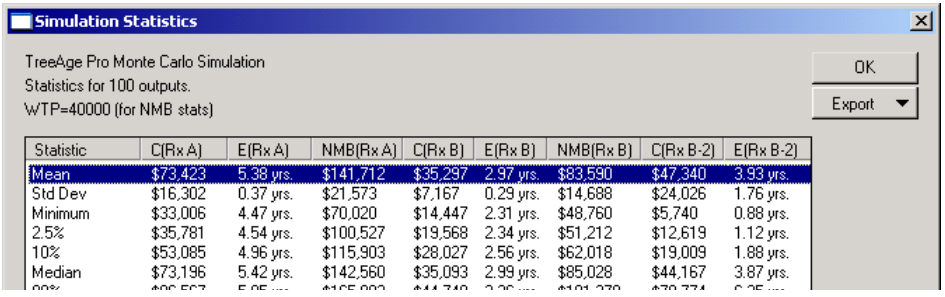
As described in Chapter 16, the Monte Carlo Simulation output window can both display running statistics while the simulation is in progress, as well as display statistics, graphs and reports once the simulation is complete.

If you halt a running simulation using the **Stop** button, TreeAge Pro can display reports and graphs based on the incomplete results. Lengthy simulations may take some time to stop after the button is pressed.



When performing a simulation in a cost-effectiveness tree, running statistics for both costs and effectiveness are displayed. When the simulation stops, this button changes to the **Stats Report…** button, which brings up the Simulation Statistics report. In a CE simulation, the report can calculate Net Benefits based on a threshold ICER.





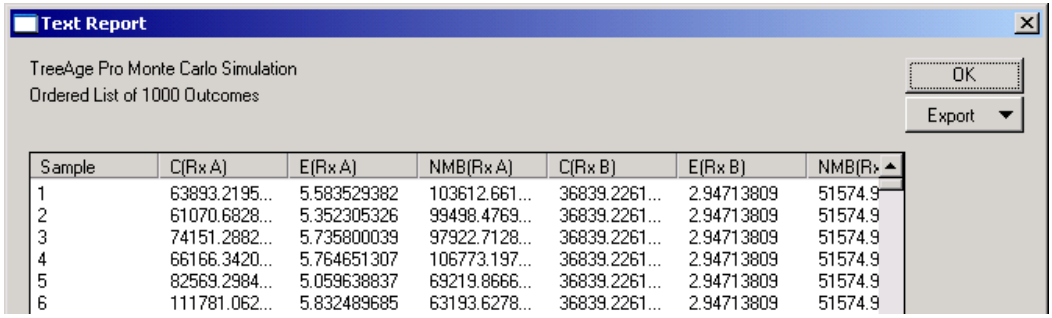
Statistic	C(Rx A)	E(Rx A)	NMB(Rx A)	C(Rx B)	E(Rx B)	NMB(Rx B)	C(Rx B-2)	E(Rx B-2)
Mean	$73,423	5.38 yrs.	$141,712	$35,297	2.97 yrs.	$83,590	$47,340	3.93 yrs.
Std Dev	$16,302	0.37 yrs.	$21,573	$7,167	0.29 yrs.	$14,688	$24,026	1.76 yrs.
Minimum	$33,006	4.47 yrs.	$70,020	$14,447	2.31 yrs.	$48,760	$5,740	0.88 yrs.
2.5%	$35,781	4.54 yrs.	$100,527	$19,568	2.34 yrs.	$51,212	$12,619	1.12 yrs.
10%	$53,085	4.96 yrs.	$115,903	$28,027	2.56 yrs.	$62,018	$19,009	1.88 yrs.
Median	$73,196	5.42 yrs.	$142,560	$35,093	2.99 yrs.	$85,028	$44,167	3.87 yrs.

Simulation text report

Once the simulation is complete, or has been stopped, the full simulation text report can be displayed. If a cost-effectiveness simulation is performed at a decision node, each option's cost is displayed in a column labeled C(*strategy*) and each option's effectiveness is displayed in a column labeled E(*strategy*). And as in the Simulation Statistics report shown on the previous page, a Net Benefits calculation can also be reported (or an average CE ratio).

Microsimulation v. PSA:

Simple microsimulation and probabilistic sensitivity analysis (PSA) have separate applications and different interpretation. Note that commonly-used PSA outputs (e.g., Net Benefits acceptability curve, ICE scatterplot) may not have an intuitive application in a microsimulation analysis. (In other words, the distribution of microsimulation outcomes is not interpreted the same way as the distribution of outputs from a PSA.)



Sample	C(Rx A)	E(Rx A)	NMB(Rx A)	C(Rx B)	E(Rx B)	NMB(Rx
1	63893.2195...	5.583529382	103612.661...	36839.2261...	2.94713809	51574.9
2	61070.6828...	5.352305326	99498.4769...	36839.2261...	2.94713809	51574.9
3	74151.2882...	5.735800039	97922.7128...	36839.2261...	2.94713809	51574.9
4	66166.3420...	5.764651307	106773.197...	36839.2261...	2.94713809	51574.9
5	82569.2984...	5.059638837	69219.8666...	36839.2261...	2.94713809	51574.9
6	111781.062...	5.832489685	63193.6278...	36839.2261...	2.94713809	51574.9

Columns are added for any distributions that are sampled, or tracker variables used in a microsimulation.

The interpretation of each value in the simulation text report will depend on the type of simulation you have performed. TreeAge Pro includes a variety of kinds of Monte Carlo simulation: one-, two-, or three-dimensional; sampling; microsimulation; and combinations of these. The Monte Carlo simulation output will report a summary for the "highest" dimension.

In the simplest case, a one-dimensional simulation, TreeAge Pro will perform a series of recalculations of the model, either expected value recalculations or individual *microsimulations* (first-order trials), and report each result. In a two-dimensional simulation, on the other hand, each row of values reported by TreeAge Pro is itself a summary of a group of recalculations of the model (again, either EV calculations or microsimulation trials).

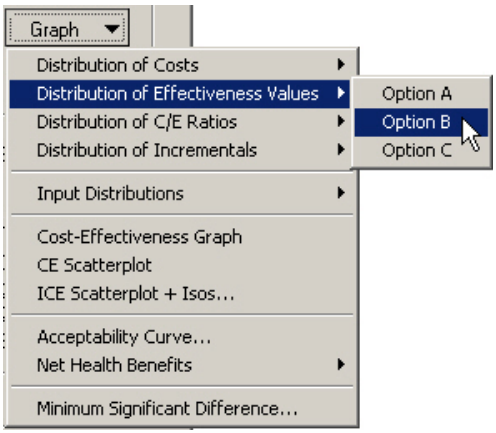
Tip:

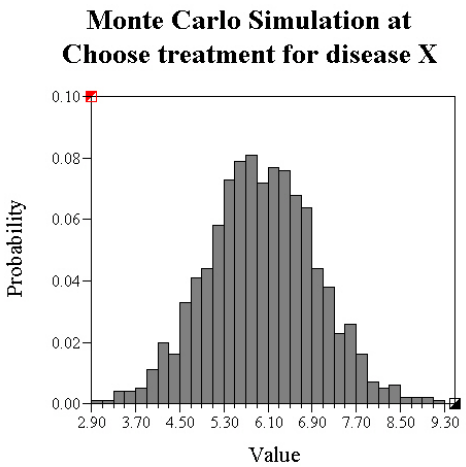
Although the simulation output window only reports the top level (or dimension) of a multi-dimensional simulation, collapsing inner levels into mean values, TreeAge Pro **does** include advanced functionality for extracting lower levels of detail. The Global( ) function, for example, can be used to automatically record and report on selected inputs and calculations. The Command() function can be used to export this information to spreadsheets.

CE graphs

The Monte Carlo simulation CE graph is the same graph as is generated by the **Analysis >** Cost-Effectiveness… command. In the simulation version, each strategy is plotted using the mean cost and effectiveness statistics from the simulation summary, rather than using expected values. The graph includes the standard CEA text report.

See the previous two chapters in this manual for detailed descriptions of the cost-effectiveness graph and text report.





Distribution graphs

In cost-effectiveness simulations, as in single-attribute simulations, a basic format for presenting the simulation results — any column in the text report — is in a probability distribution histogram. Each Distribution histogram includes a statistical summary report, in addition to a full text report. See Chapters 5 and 6 for complete details on using and customizing distribution histograms in TreeAge Pro.

In cost-effectiveness simulations performed at decision nodes, distributions are available not only for each strategy's cost and effectiveness columns, but for incremental values as well. Incremental distribution graphs compare one strategy to a baseline strategy.

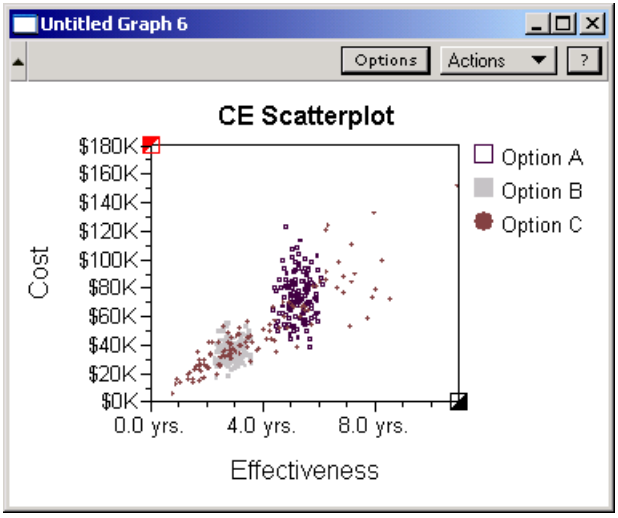
Although there is natural interest in using the distribution histogram of ICERs to visualize the uncertainty about the cost-effectiveness of strategies, considerable care must be taken in using this graph. The same issues described in the previous chapter as complicating the interpretation of ICER thresholds in sensitivity analysis graphs, also applies in Monte Carlo simulation. It is critical that the scatterplots, acceptability curves, and net benefits graphs and histograms described below be used to put the ICER distribution graph in context — in particular, to determine whether there is uncertainty about the sign (positive or negative) of incremental effectiveness.

In addition to reading about the other types of CE simulation graphs and reports below, read more about the issues surrounding uncertainty analysis and incremental CE ratios, either in one of the references given at the beginning of Chapter 32, or in journal publications such as:

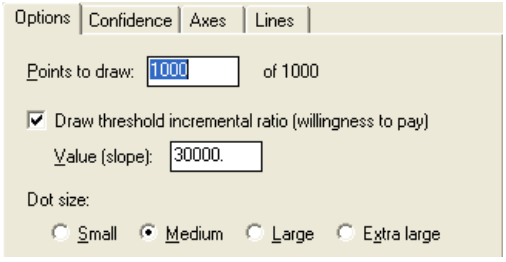
> "Quantifying stochastic uncertainty," Glick, Briggs, and Polsky, *Expert Rev Pharmacoeconomic Out Res 1(1), 25-36 (2001)* [and www.future-drugs.com].

Scatterplots

The CE graph, which plots the mean cost and mean effectiveness of each strategy, can be naturally extended to a scatterplot for the simulation. The CE scatterplot uses the cost-effectiveness plane to plot the individual cost and effectiveness pairs for each recalculation of the model. If the simulation is performed at a decision node, each strategy's set of points uses a different color.



Depending on the number of iterations included in the simulation, it may be useful either to include only a subset of results in the plot (e.g., if the general density of points is too high) or to increase the size of the dots used in the plot (e.g., if there are only a few points to display).



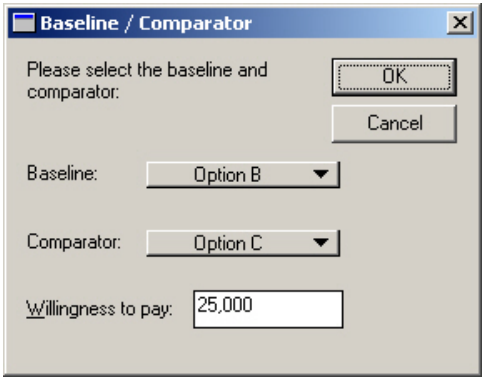
Both of these settings can be modified using the Graph Options dialog, which can be opened by clicking the Options button in the graph window. To change dot color or shape, double-click on the marker in the graph legend.

See Chapter 6 for more on graph options.

The ICE scatterplot has a variety of other features and options, described on the following pages.

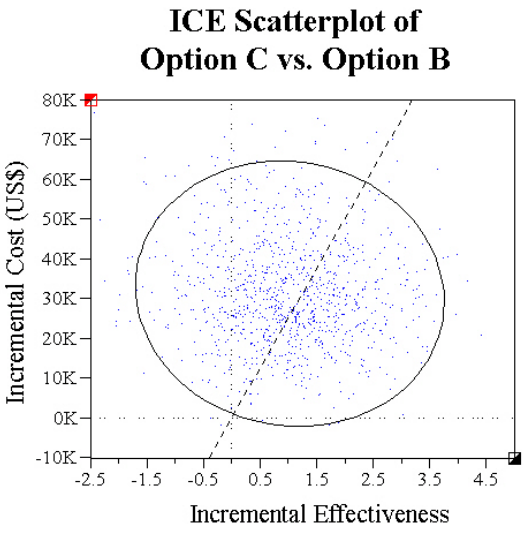
The Incremental Cost-Effectiveness (ICE) scatterplot

Like the CE scatterplot, the ICE scatterplot uses a form of the standard CE plane to plot points for each iteration in the simulation output. The ICE scatterplot includes a single set of points representing pairs of incremental cost and effectiveness values from the simulation results, based on a comparator (e.g., Option C) relative to a baseline (e.g., Option B). In the setup dialog for the ICE scatterplot, you must select the comparator and baseline strategies.

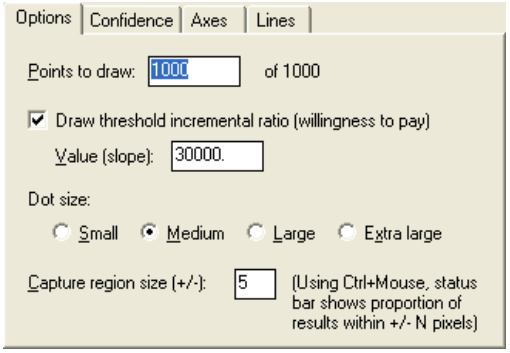


Generally, if an option has a higher mean cost and effectiveness in the simulation statistical summary, it should be specified as the comparator. The points in the scatterplot will represent the comparator's incremental cost and incremental effectiveness relative to the baseline (represented by the origin).

TreeAge Pro also prompts for the WTP, or ceiling ICER, to use as the slope of a line intersecting the origin of the plot. The WTP line in the graph intersects points having the specified ICER value, and the region below the line includes cost-effective points. This is utilized in the scatterplot's text report to calculate the percentage of simulation iterations for which the comparator is cost-effective. (A similar analysis is done using a different graph, the Acceptability Curve, which works for all strategies in the analysis, not just two.)

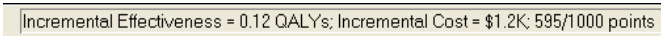


The WTP value can be changed after the graph is displayed, via the Options dialog.

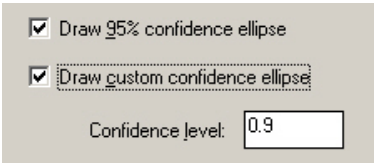


In addition to the WTP line and confidence ellipses, a number of other settings in the plot can be changed using the Graph Options dialog, including the number of points plotted, the size of the points, and the special cursor capture region.

The Capture region setting controls the size of a special cursor in the graph. When you hold down the control key, the mouse cursor will report in the status bar how many iterations fall under the cursor (e.g., stacked on the origin, or an axis).



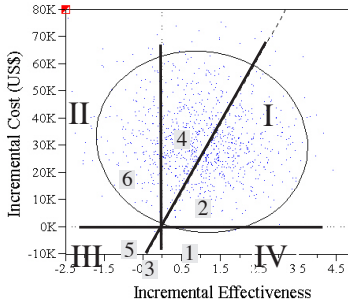
In addition to the WTP line, TreeAge Pro draws a 95% confidence ellipse in the ICE scatterplot. The display of this confidence ellipse can be turned off or on in the scatterplot's Options dialog, under the Confidence tab. It is also possible to activate a second confidence ellipse using a confidence level percentage other than 95%.



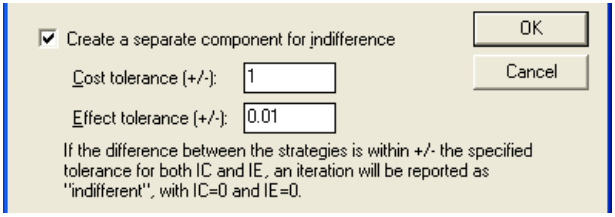
Tip:

For details on how confidence ellipses account for correlation between cost and effects, see for example "Reflecting Uncertainty in Cost-Effectiveness Analysis", Manning et al, Ch. 8 in *Cost-Effectiveness in Health and Medicine*, Gold et al., Oxford Univ. Press (1996).

The threshold ICER (WTP) drawn in the graph is also the basis for the text report's calculation of the probability of fall within the six component regions of the ICE scatterplot, as illustrated below.



An additional region of indifference can be created, capturing points on or near the origin.


Create a separate component for indifference
OK
Cancel
Cost tolerance (+/-):
1
Effect tolerance (+/-):
0.01
If the difference between the strategies is within +/- the specified tolerance for both IC and IE, an iteration will be reported as "indifferent", with IC=0 and IE=0.

For each plotted result, the comparator will be cost-effective (relative to the baseline) if it falls within the three component regions *below* the threshold line (C1–C3).

The text report rows corresponding to each component are as follows:

C1 — comparator **dominates** the baseline

C2 — comparator is more costly and effective, and its ICER is less than or equal to the WTP, so it is **cost-effective**

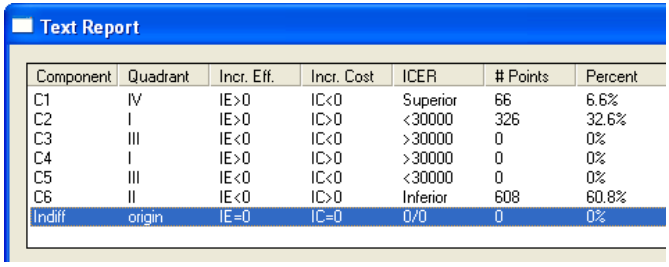
C3 — baseline is more costly and effective, but its ICER is greater than the WTP, so the comparator is **optimal**

C4 — comparator is more costly and effective, but its ICER is greater than the WTP, so the baseline is optimal

C5 — baseline is more costly and effective, and its ICER is less than or equal to the WTP, so it is optimal

C6 — baseline dominates the comparator

Indiff — incremental C and E close to zero



Component	Quadrant	Incr. Eff.	Incr. Cost	ICER	# Points	Percent
C1	IV	IE>0	IC<0	Superior	66	6.6%
C2	I	IE>0	IC>0	<30000	326	32.6%
C3	III	IE<0	IC<0	>30000	0	0%
C4	I	IE>0	IC>0	>30000	0	0%
C5	III	IE<0	IC<0	<30000	0	0%
C6	II	IE<0	IC>0	Inferior	608	60.8%
Indiff	origin	IE=0	IC=0	0/0	0	0%

Increasing the WTP value changes the shape of the component regions #2 – 5, and thus the report. See the section on Acceptability Curves below for a robust method of testing a range of WTP values for all potentially cost-effective strategies.

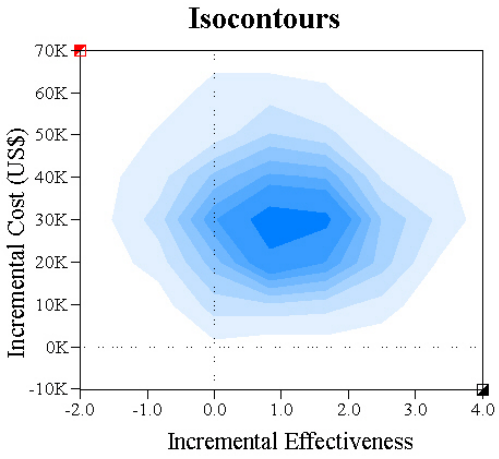
For additional discussion, refer to "Uncertainty in Decision Models Analyzing Cost-Effectiveness," Maria Hunink et al, *Med Decis Making* 18:337-346 (1998).

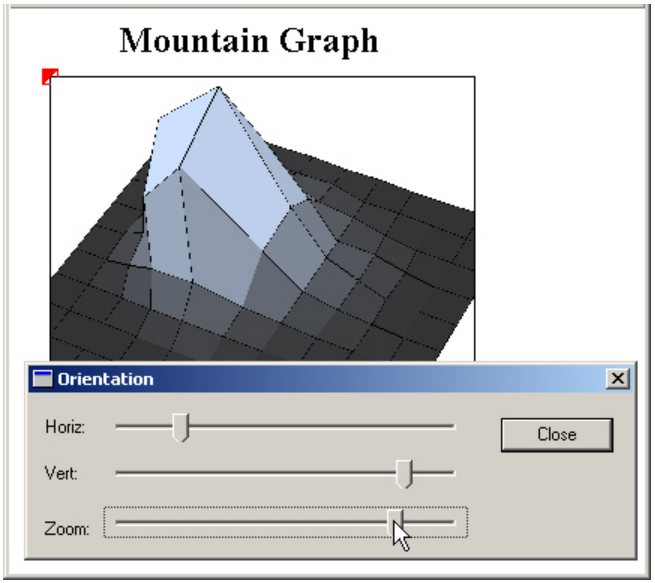
ICE density plots: isocontours and mountain graph

The ICE scatterplot can be converted into two related graphs, an isocontour graph or a 3-D mountain graph, using the **Convert To** button at the top of the graph window.



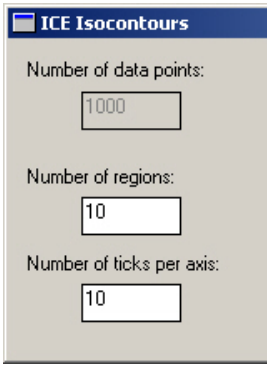
The ICE isocontours graph shows the relative concentration of points in the scatterplot, using a range of colors/shades to indicate regions of different concentration. The mountain graph is a 3-dimensional representation of the isocontour graph, indicating different concentrations of points using both color/shade and a virtual third dimension, height. In both the isocontour and mountain graphs, regions are created using simple point-counting methods.



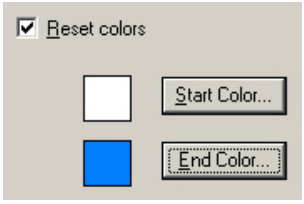


Prior to converting the scatterplot, a dialog is displayed in which conversion settings are specified. In the isocontour graph setup dialog, shown on the next page, there are three settings that determine: 1) the number of different regions of point density to draw; 2) the granularity or smoothness of the outlines of each isocontour; and 3) the number of points to use from the ICE scatterplot's data set. The 3-D mountain graph's setup dialog is similar, except that it only prompts for the granularity setting and the number of data points to use.

The higher the values used for these initial settings, the more detailed the resulting graphs will be. However, these settings are limited by the number of points in the original data set (e.g., the number of samples or trials in the simulation). In general, the default settings can be accepted, initially. Click OK in the setup dialog to display the graph using the specified conversion settings.



Once the two-dimensional isocontours graph is displayed, you can click on the Options button and use the Colors tab to modify the color range used.



In the 3-D mountain graph, click on the Orientation button to display the three rotation controls used to rotate the graph on any of its three axes.

For additional discussion of these variations on the ICE scatterplot, refer to "Uncertainty in Decision Models Analyzing Cost-Effectiveness," Maria Hunink, Jan Roelf Bult, Jelle de Vries, and Milton Weinstein, *Med Decis Making* 18:337-346 (1998).

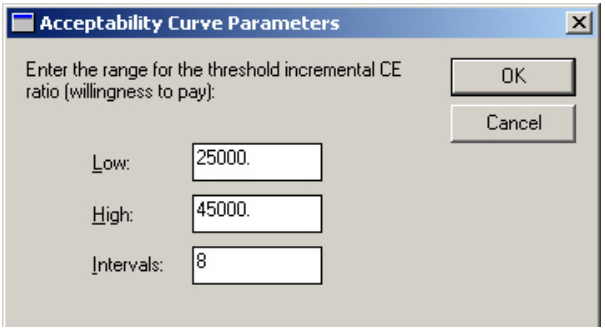
Acceptability and Net Benefits curves

The acceptability curve is a commonly-used visual aid for communicating the results of probabilistic sensitivity analysis in cost-effectiveness models.

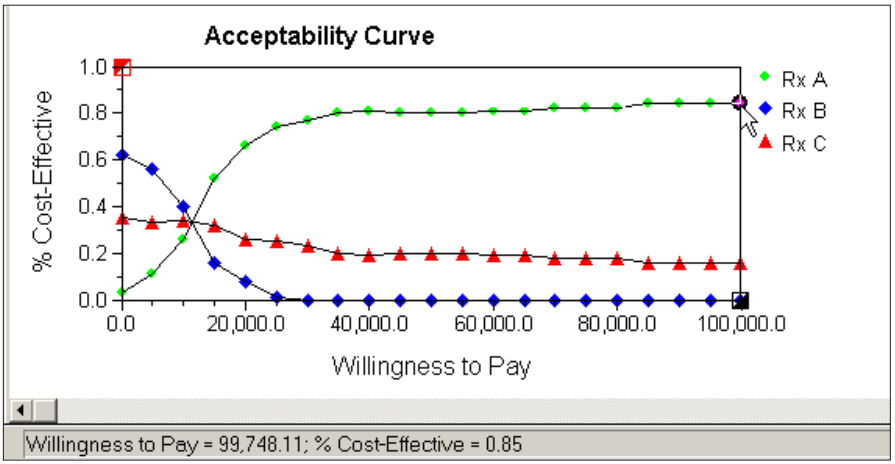
TreeAge Pro includes two forms of acceptability curve, both presenting relative cost-effectiveness as a function of the threshold ICER. The *pairwise* version uses ICERs to graph the changing percentage of iterations for which a comparator is cost-effective relative to a single baseline strategy.

A second, more powerful form uses *net benefits* (see Chapter 32, and below) to graph the changing percentage of iterations for which each comparator is cost-effective relative to all other strategies. The net benefits acceptability curve is extremely useful when there are more than two strategies under consideration.

Like a sensitivity analysis, the acceptability curve requires a range of values for the threshold ICER, as well as the number of intervals into which the range should be divided. (If you select the pair-wise format, TreeAge Pro will prompt you to pick the baseline option.) After setting the appropriate parameters, click OK to display the graph.



The net benefits version of the acceptability curve includes a line for each strategy, with the curves summing to 1.0 (100%) at each interval.



The interpretation usually applied to the net benefits acceptability curves is that the graphed value of any comparator at a particular WTP represents the probability that it is cost-effective (most effective option within the threshold ICER), based on the uncertainties included in the simulation.

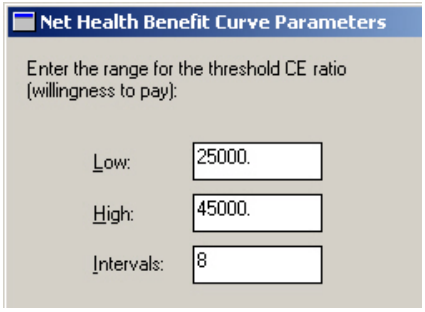
The value of a comparator at WTP=0 represents the probability that it is the least costly option; the *limit* of a comparator's line as it is followed to the right (approximated in the graph above by control-clicking on the line) represents the probability that it is the most effective option.

For more information on the net benefits acceptability curve, refer to:

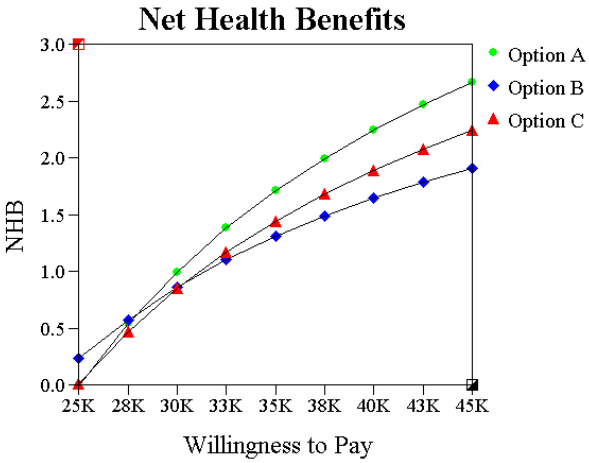
> "Quantifying stochastic uncertainty" Glick, Briggs, and Polsky, *Expert Rev Pharmacoeconomic Out Res 1(1), 25-36 (2001)* [and www.future-drugs.com].

Net Benefits vs. WTP

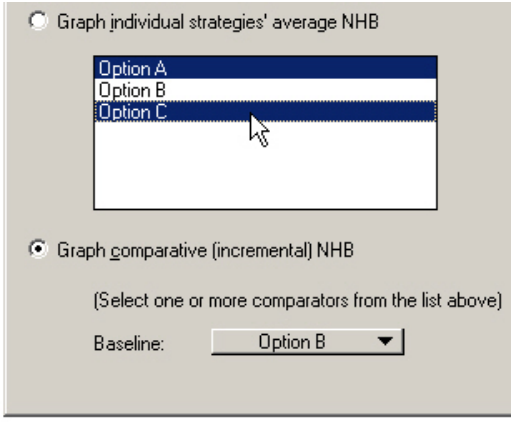
Net health benefits (NHB) and net monetary benefits (NMB) calculations integrate a particular WTP value, as explained in Chapter 32, and so will prompt you to furnish either a WTP value or value range. The Net Benefits vs. WTP graph, in both average and incremental formats, functions as a sensitivity analysis on WTP.

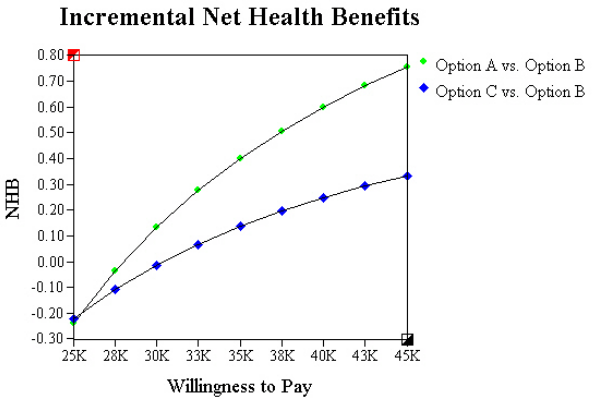


An intervention's mean effectiveness and cost statistics for the simulation are the only other inputs for each calculation of that intervention's net benefit.



To graph incremental net benefit curves, select the second radio button, use the pop-up menu at the bottom of the dialog to choose the baseline option, and then choose one or more other comparators from the list box above. Using the same baseline, a curve of incremental net benefit as a function of WTP will be drawn in the graph for each selected comparator.





Incremental net health benefit (INHB) is calculated as:

$$INHB_{C-B} = NHB_C - NHB_B$$

where C refers to a comparator and B refers to the baseline. An alternative form of the equation, providing the same result, is:

$$INHB_{C-B} = (E_C - E_B) - ((C_C - C_B) / WTP)$$

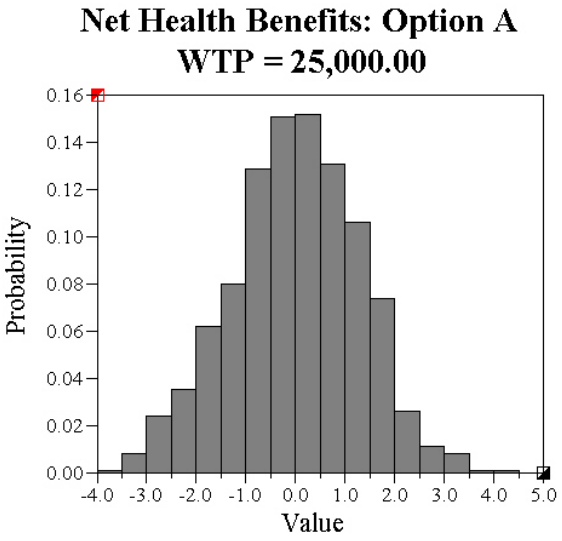
Net benefits probability distributions

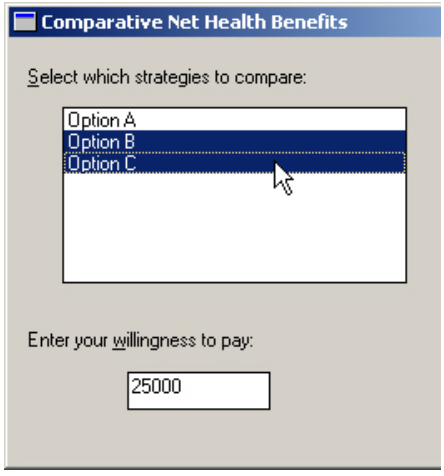
Histograms of the distribution of net benefit values for a single option in a cost-effectiveness simulation are generated based on a single WTP value. Before generating a histogram, TreeAge Pro will prompt you to enter the WTP.

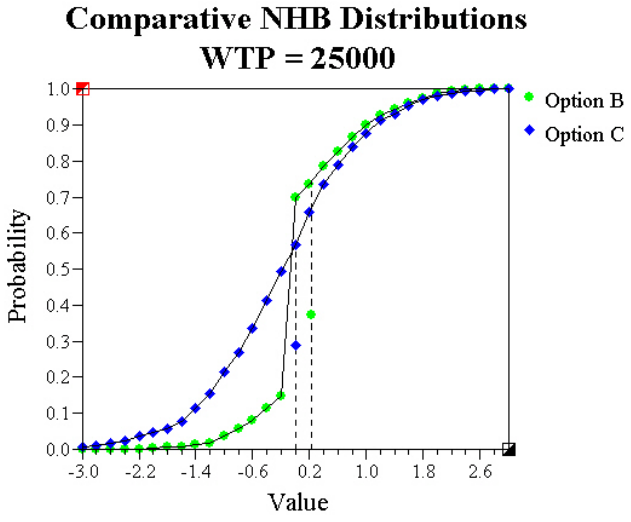
As with other distributions, after generating the graph, you can identify statistics such as the mean and confidence intervals for a particular histogram by clicking on the Statistics button at the top of the graph window.

A Net Benefits histogram can be changed to its cumulative form, simply by clicking on the Options button and checking the option labeled Cumulative.

If you choose the comparative version of the Net Benefits probability distribution from the simulation Graph button menu, TreeAge Pro prompts for a WTP value and a list of options to include in the graph. This graph uses, by default, a continuous form of the cumulative probability distribution histogram to facilitate displaying on a single graph the distribution of Net Benefits values for multiple comparators.







The comparative Net Benefits graphs, like the expected value version of the comparative probability distribution graph described in Chapter 5, is designed to enable evaluation of conditions of stochastic dominance between strategies. In the net benefit framework, strong stochastic dominance is said to be present if the dominant strategy's Net Benefits curve always falls below that of the dominated strategy (although the lines may touch).

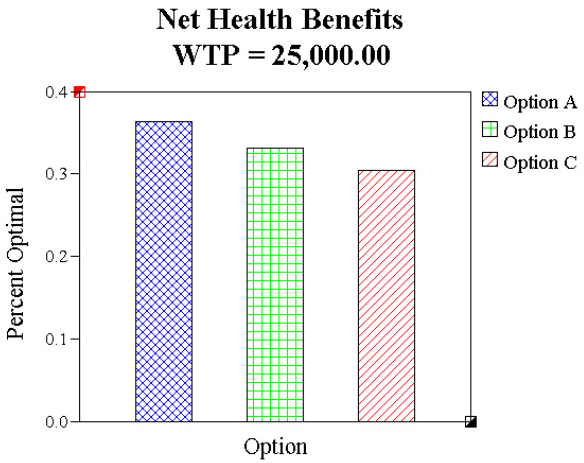
If you select the incremental form of the histogram, in addition to specifying the WTP, you must also select one baseline strategy and one comparator strategy. Based on the simulation trials or samples, the probability distribution of the comparator's incremental Net Benefits will be graphed.

As with other probability distribution histograms, the Options dialog includes a check box to display the histogram in cumulative form.

For background and additional theoretical discussion of Net Benefits graphs and dominance, refer to "Net Health Benefits," Aaron Stinnett and John Mullahy, *Med Decis Making* Suppl. 18:S68-S80 (1998).

Optimality graph

Based on a single WTP value, the Net Benefits optimality graph displays for each option the percentage of trials or samples in a cost-effectiveness simulation for which that option is optimal. Optimality is determined simply by selecting the option with the highest net benefit for that trial/sample. See Chapter 32 for details.



Cost-effectiveness value of information (EVPI and EVPPI)

As described in Chapter 16, using the results of a Monte Carlo simulation performed at a decision node, TreeAge Pro can calculate the expected value of perfect information (EVPI) or partial EVPI, based on hypothetically eliminating the simulated uncertainties (normally, one or more sampling distributions).

Text Report

TreeAge Pro Monte Carlo Simulation
Ordered List of 102 Outcomes

Sample	IC with PI	IE with PI	EVPI (INMB)	Cost with PI ...
Expected Val...	-5905.51...	0.15074...	10427.964...	67517.2574...
Samples				
1	0	0	0	60000.7946...
2	0	0	0	45175.1702...
3	-10035.3...	0.65702...	29746.076...	86531.7755...
4	900.889...	2.23445...	66132.867...	76102.0869...
5	0	0	0	71585.5741...

Notes:

** = Single best strategy overall, based on expected values
EVPI = 'Expected Value of Perfect Information'
PI = 'Perfect Information'
Specified threshold ICER = 30k
NMB = 'Net Monetary Benefit' = (thresholdICER*eff - cost)

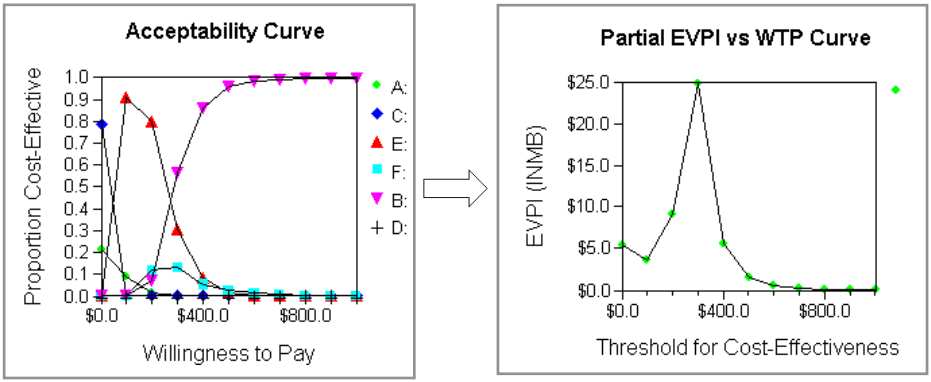
A detailed **EVPI report**, available under the **Graph** menu, shows the calculations for a single WTP. In a cost-effectiveness simulation, the report is based on Net Benefits calculations. TreeAge Pro prompts for a WTP to do the Net Benefits calculations.

If the optimal strategy for one iteration is not the optimal strategy overall, then there is some value to eliminating the uncertainties being simulated. The average, or expected, value of information for all iterations is reported in the top row of the report. In the example at left, the expected cost savings from perfect information (per patient) is 5900 and the expected gain in effectiveness is 0.15.

EVPI vs. WTP Curve

TreeAge Pro can also generate a curve reporting the changing EVPI (in terms of Net Benefit) as the WTP is varied. The resulting EVPI curve looks like the inverse of the acceptability curve.

In the EVPI curve and corresponding acceptability curve shown below (based on a different analysis than the report above), when there is significant uncertainty about cost-effectiveness (e.g., given a particular WTP, 45% of samples prefer "B" while 45% prefer "E") there is a higher value of information.



Markov Modeling, Analysis, and Microsimulation with the Healthcare Module

The chapters in Part VIII of the manual provide information on creating and analyzing Markov models using the Healthcare module.

Advanced Markov modeling and Monte Carlo microsimulation features are also covered in detail.

In this part:

- Chapter 34, "Building and Analyzing Markov Models"
- Chapter 35, "Advanced Markov Modeling Techniques"
- Chapter 36, "Markov Monte Carlo Microsimulation"

CHAPTER 34

BUILDING AND ANALYZING MARKOV MODELS

This chapter covers the basics of creating and analyzing Markov processes with the TreeAge Pro Healthcare module. Some basic conceptual background is provided.

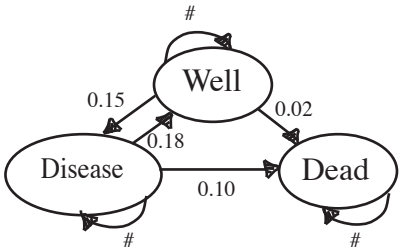
Chapter 35 covers a variety of important Markov topics including time-dependent probabilities, discounting, half-cycle correction, tunnel states, and Markov decision processes. Markov microsimulation is covered in detail in Chapter 36.

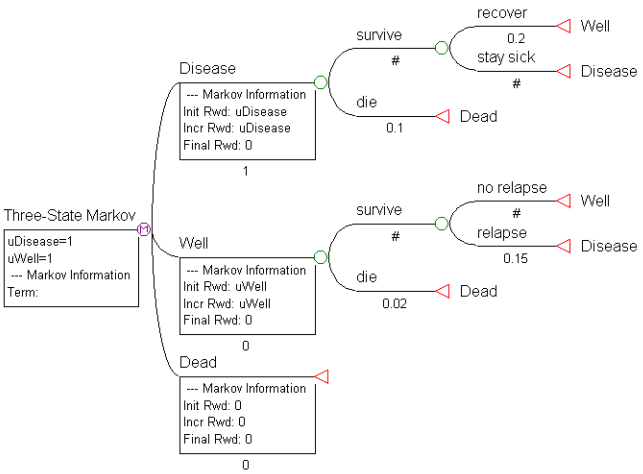
In this chapter:

- Markov modeling basics..... 446
- Building a Markov cycle tree in TreeAge Pro 448
- Analyzing a Markov model..... 458
- A note on microsimulation..... 465
- A note on half-cycle correction..... 465
- Cost-effectiveness Markov models 466

Markov modeling basics

While most decision trees include a simple notion of time (i.e., events on the left side of the tree occur after those on the right), there are no shortcuts in a standard tree structure for representing events that recur over time. A state transition model, also called a Markov model, is designed to do just this. Markov models are used to simulate both short-term processes (e.g., development of a tumor) and long-term processes (e.g., an individual's lifespan).





State transition models

Markov models built in TreeAge Pro often represent discrete-time state transition models (although *discrete event* modeling is also possible). A discrete-time Markov model usually follows a basic design, such that:

- The time period of interest (i.e., 10 years) is divided into equal intervals, or *cycles*.
- A finite set of mutually exclusive *states* is defined such that, in any given cycle, a member of the cohort is in only one state.
- *Initial probabilities* determine the distribution of cohort members among the possible states at the start of the process (often, the entire cohort starts in the same state).
- A matrix of *transition probabilities*, applied in each successive cycle, defines the possible changes in state.
- To calculate an expected value for the model, (e.g., net cost or quality-adjusted life expectancy), different cost and/or utility *rewards/tolls* are accumulated for each interval spent in a particular state.

Graphical representation

In a bubble diagram, like that shown above left, each state is represented using an oval, arrows represent transitions, and numbers along the arrows indicate the transition probabilities. The probabilities of the transition arrows emanating from any state must sum to 1.0.

TreeAge Pro does not employ the bubble diagram representation of a Markov model. Instead, TreeAge Pro uses a graphical form known as a *cycle tree*, which is more flexible and easily integrated into decision trees. Markov cycle trees can be appended to paths in a TreeAge Pro decision tree anywhere you might place a terminal node.

Calculation basics

There are two commonly-used methods for evaluating a Markov model: expected value calculation (called "cohort" analysis), and Monte Carlo microsimulation (first-order trials). It is important to understand the difference between the two analysis methods, and to recognize the terms associated with them.

In an expected value analysis, the *percentage of a hypothetical cohort* in a state during a cycle is multiplied by the cost or utility associated with that state, and these products are summed over all states and all cycles. In TreeAge Pro, expected value calculations are the basis of most analyses, including *n*-way sensitivity analysis, and baseline cost-effectiveness analysis.

In a *microsimulation*, on the other hand, a single trial's value is simply the sum of the rewards or tolls from the states traversed by an "individual" taking a random walk through the model based on transition probabilities. An expected value is estimated by averaging many such trials.

In TreeAge Pro, the same Markov model can be evaluated by either expected value or microsimulation methods. Generally, deterministic, expected value analysis is preferred because it is more computationally efficient; it returns a mean value much more quickly than simulation, which often requires thousands of trials to return a mean value within an acceptable error.

Additional background discussion can be found in:

- *Decision Making in Health and Medicine,* Hunink, and Glasziou (2001), Cambridge University.

You are urged to consult these and other publications dealing with the concepts which underlie Markov modeling.

Non-standard Markov models

In TreeAge Pro, the basic Markov modeling rules outlined above can be overruled in a variety of ways, for example:

- Time-dependent Markov models are easily handled using tables, tunnels, and/or tracker variables.
- Discrete event models can combine sampling from event time distributions and microsimulation features like tracker variables.
- A Markov model can be analyzed using the Node() function in such as way that sensitivity analysis and other cohort-type analyses can be used, while the Markov model is actually evaluated using microsimulation trials.
- EV/cohort analysis of a Markov model can make use of a realistic cohort with a specific starting size and composition that may change over time.

These kinds of features are covered in Chapters 35—37.

Building a Markov cycle tree in TreeAge Pro

The design of a basic Markov model requires consideration of a number of components, most of which have been introduced above:

- **States** – The set of distinct health states under consideration in the model, together with the possible transitions between them.
- **Cycle length** – The length of time represented by a single stage (or cycle) in a Markov process. This value is implicit in the probabilities, rewards/tolls, and termination condition
- **Initial probabilities** – A set of probabilities used only at the outset of the process, describing the initial distribution of the cohort among the states.
- **Transition probabilities** – The matrix of probabilities of moving between health states from one stage to the next.
- **Rewards/tolls** – Per-cycle costs and/or utilities (corresponding to payoffs in regular trees) representing the outcome measure(s) being calculated, e.g., costs or QALYs. Rewards/tolls may be associated with health states, or with instantaneous, short-term events.
- **Termination condition** – A logical test evaluated at the beginning of each new cycle to determine if the process should continue or stop.

These elements will be illustrated using the simple, three-state Markov state transition model illustrated at the beginning of the chapter.

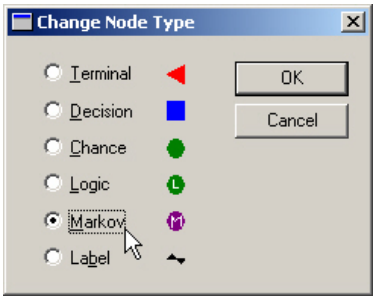
The Markov node

To begin, a Markov node must be used. Any number of Markov nodes can be included in a decision tree. In this case, the root node of an empty tree will be changed to a Markov node.

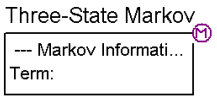
► **To create a Markov node:**

▷ Create a new tree and select the root node (or you can use the right-most node in any scenario in a tree, other than a node within an existing Markov subtree).

▷ Choose **Options >** Change Node Type..., click on Markov, and press ENTER or click OK.



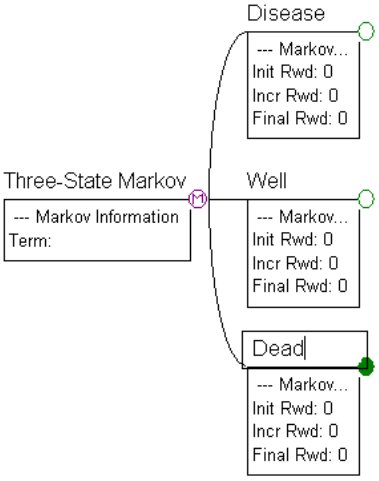
▷ Type `Three-State Markov` for a text description of the new Markov node.



Note that, in addition to using the purple Markov node symbol, TreeAge Pro also adds a Markov information box below the node; we will return to this later.

Markov states (and initial probabilities)

The branches of the Markov node enumerate the Markov states, and are labeled as such. For our simple three-state model, we need three branches from the Markov node. The standard tree-building tools found under the Options menu, and covered in the Part I of the manual (Change Node Type, Add Branches, Insert Branch, Copy/Paste Subtree, etc.) are used to create the Markov subtree.



► **To add the Markov states:**

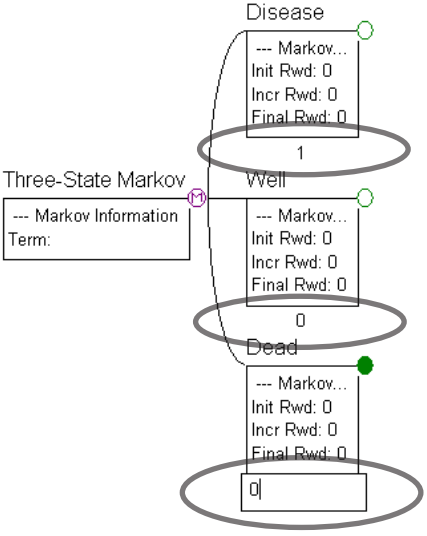
▷ Select the Markov node, and add three branches.

▷ Type the labels `Disease`, `Well`, and `Dead` above the new branches.

TreeAge Pro's use of arcs, rather than straight lines, for the branches within a Markov subtree is simply to make it easier to distinguish Markov subtrees from the rest of a decision tree.

Below the Markov state branches, *initial state probabilities* must be entered. These probabilities, like the probabilities of a chance node, must sum to one. It is often the case that many states will have initial probabilities of 0. In the three-state model, for example, 100% of the cohort begins in the Disease state.

▷ Below the branches of the Disease, Well, and Dead states, enter the initial probabilities, `1`, `0`, and `0`, respectively.



Initial state probabilities are used by TreeAge Pro only once during the Markov process, to determine where individuals should spend the first cycle of the process. All subsequent movement through the model utilizes *transition probabilities*, which you will specify later.

State rewards

In TreeAge Pro, costs or utilities assigned in a Markov model are called *rewards*. A *state reward* refers to a value that is assigned to individuals because they spend one cycle in a particular state. This might be a cost or a unit of life expectancy, for example.

The state rewards must reflect the *length of a cycle*. For instance, if you have decided on a yearly cost of $6000 for a particular state, but your cycle length is 1 month (not 1 year), then the state reward (cost) should be 6000/12, or 500 per cycle.

Assume that the goal is to estimate average life expectancy, and that the model's cycle length is one year. To calculate life expectancy in terms of years, you would assign a state reward of 1 to any alive state. (If the cycle length were 1 month, a state reward equal to 1/12 would be used to calculate life expectancy in years.)

Note on regional (European) numeric settings:

If your computer is set up to use commas (",") to represent decimals, rather than periods ("."), you should enter numbers in TreeAge Pro in this fashion, just as you would in a spreadsheet or calculator. (If you need to setup a tree for use under different regional numeric settings, TreeAge Pro also has preference settings that enable a tree to override a computer's regional settings or reverse the usage of decimals; see Chapter 12 for details.

► **To define state rewards:**

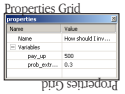
▷ Select the Disease node and choose **Values >** Markov State Information... (or click on the Markov tool bar button, or use the right-click menu).



▷ In the Markov State Info dialog, shown on the following page, the Rewards pop-up menu should be set to 1 (equivalent to Payoff #1 in a terminal node).

▷ If the properties grid is visible, it can be used to enter or modify the state rewards for the currently selected state. Click on the plus (+) sign labeled "Markov Information" to show the three state reward expressions.

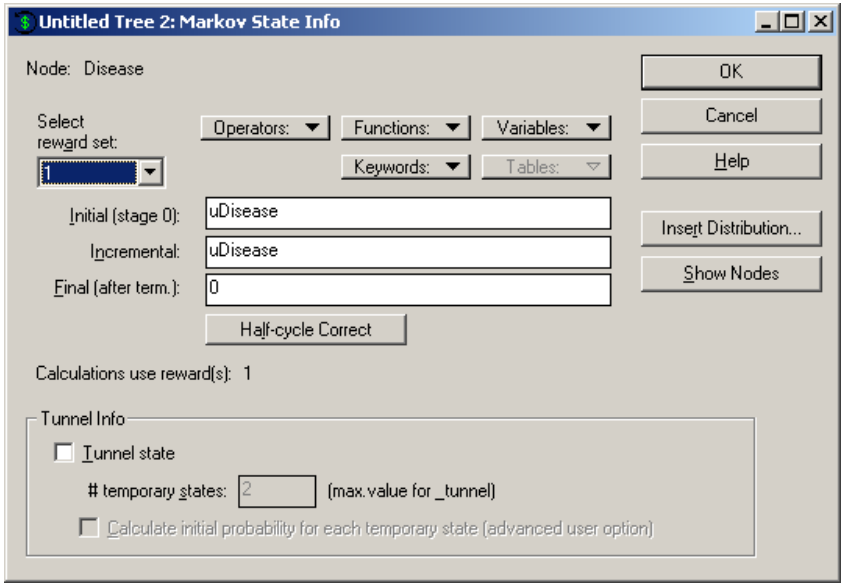
Properties Grid



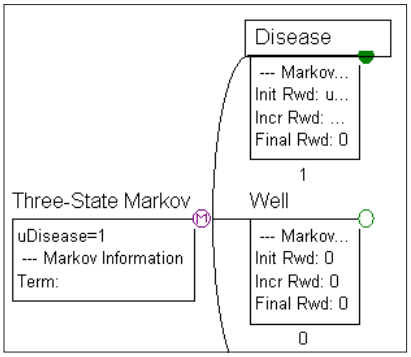
For life expectancy (or other) calculations, you can enter three separate state reward expressions at each Markov state. The reasons for having three separate state rewards — for the half-cycle correction, primarily — will be explained in detail in the next chapter.

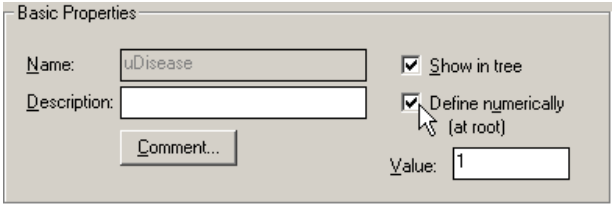
A state's *initial* reward is assigned **only in the first cycle**, stage 0, and only to individuals that spend stage 0 in that state. The *incremental* reward is assigned in subsequent cycles during the process. The *final* reward (if any) is assigned after the process is over to individuals ending up in that state.

In this case, we want to assign a reward of 1 for each year/cycle someone spends in an alive state. Instead of assigning numeric initial (stage 0) and incremental state rewards of 1 for the Disease and Well states, use a variable to represent the "utility" of each state.



- In the text boxes labeled Initial, Incremental, and Final, enter rewards of `uDisease`, `uDisease`, and `0`, respectively.
- Press ENTER OR CLICK OK.
- TreeAge Pro will validate the reward expressions you entered, and prompt you to create the variable, uDisease. In the variable's Properties dialog, define it numerically (at the root node) equal to 1.





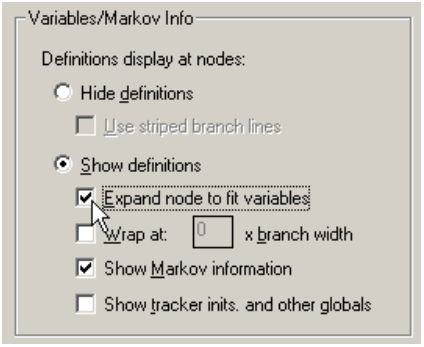
TreeAge Pro displays the new definition of uDisease at the root, Markov node, in the same box as the Markov information.

Only part of the state reward text is displayed because, by default, text in the definitions box is truncated to the width of the node label. To fix this, either type a wider node name, or change the tree's display preferences.

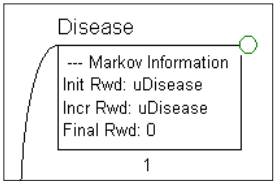
► **To change the Markov information display settings:**

▷ Choose **Edit >** Preferences.

▷ In the Preferences dialog, select the Variables/Markov Display category. Check the option labeled Expand node to fit… and click OK to update the tree.



Nodes with definitions will now show the entire text.



The Show Markov information preference can be turned off, to completely hide the Markov information, if desired, to display a more compact view of the tree for printing.

Now enter the state rewards for the other alive Markov state, Well.

▷ Select the Well state and choose **Values >** Markov State Information....

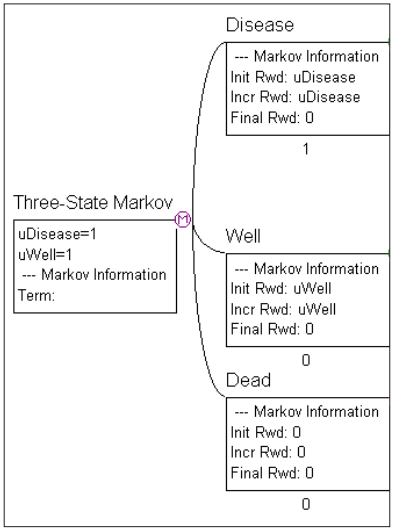
▷ For the initial, incremental, and final rewards, enter `uWell`, `uWell`, and `0`, respectively. Press ENTER or click OK to return to the tree.

▷ TreeAge Pro will prompt you to create the variable uWell. Define the variable numerically equal to 1.

Technically, assigning an initial reward in Well is unnecessary, because its initial probability is 0. It does not hurt to specify it, however, as this gives you the flexibility of later changing the initial probability of Well, for instance in a sensitivity analysis. Similarly, using variables for the rewards provides flexibility for later modifications.

The zero rewards can be left unchanged in the Dead state.

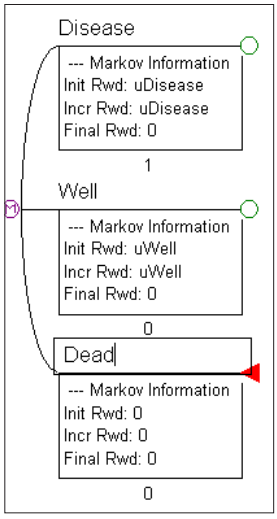
▷ Save the partially complete tree as Three-State Markov.



Transition subtrees and absorbing states

The branches of the Markov node represent the possible states. The branches (or subtree) emanating from a Markov *state*, on the other hand, represent possible events during a cycle in that state, including transitions to other states.

The easiest state to complete is Dead. A state from which an individual cannot exit has no transition subtree, and is called an *absorbing* state. To represent an absorbing state, a Markov state is simply changed to a terminal node. Markov models are not required to have any absorbing states. In this example, as in often the case, the Dead state is the sole absorbing state.



► **To create an absorbing state:**

▷ Select the Dead state, choose **Options >** Change Node Type…, and change it from a chance node to a terminal node.

Unlike a Markov bubble diagram, Markov cycle trees that you create in TreeAge Pro can represent a series of events that can occur during a single cycle. Any number of chance nodes, as well as logic and label nodes, can be used to the right of a Markov state.

To illustrate this in the example, the transition subtrees for the Disease and Well states each use two chance nodes (instead of one chance node with three branches). One chance node represents mortality (including excess mortality in the Disease state), and a second represents whether or not an individual gets/stays sick.

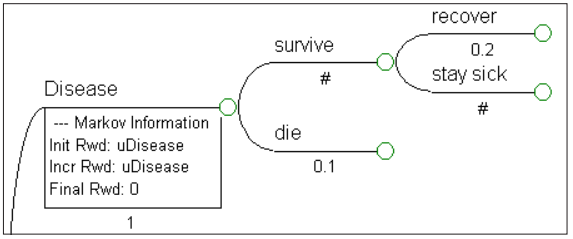
► **To create a transition subtree:**

▷ Select the Disease state chance node, and add two branches.

▷ Type the labels `survive` and `die` above the Disease node's two branches; below the branches, enter the probabilities # and `0.1`, respectively.

▷ Add two branches to the new, survive chance node.

▷ Type the labels `recover` and `stay sick` above the survive node's two branches; below the branches, enter the probabilities `0.2` and #, respectively.

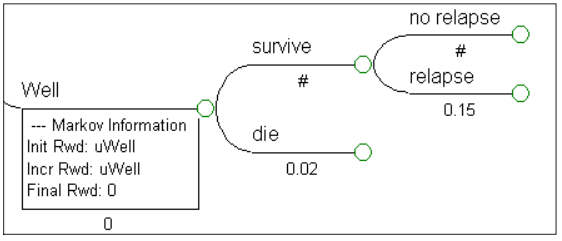


Now, create the Well state's transition subtree.

▷ Select the Well state chance node, and add two branches.

▷ Type the labels `survive` and `die` above the two branches; below the branches, enter the probabilities # and `0.02`, respectively.

▷ Add two branches to the new, survive chance node. Type the labels `no relapse` and `relapse` above the branches; below the branches, enter the probabilities # and `0.15`, respectively.



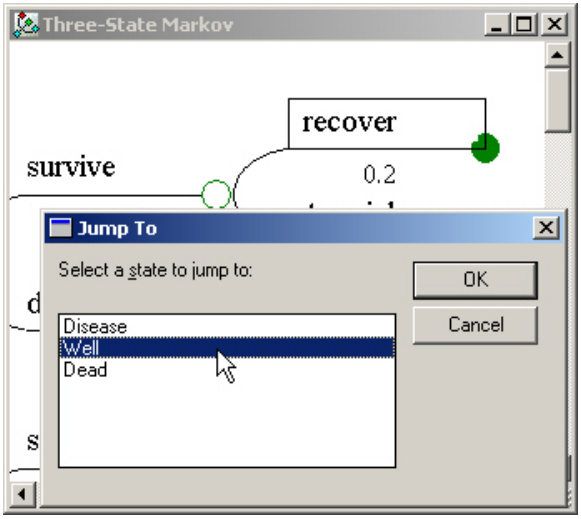
All that is left to do is to terminate each path in the transition subtrees. These *transition* nodes represent the last event in each path during a cycle, not the end of the Markov process. Individuals reaching a transition node are pointed to a Markov state where they will begin the next cycle (if the process is not terminated first).

► **To create jump nodes:**

▷ Select the recover node in the Disease state's transition subtree, and change it to a terminal node using the **Options >** Change Node Type… command.

▷ TreeAge Pro will automatically open the Jump To dialog. In the dialog, select Well from the list of existing states as the appropriate jump-to state for the recover node, and or click OK.

▷ Now, change the stay sick node to a terminal node, and have it jump to Disease. Change the die node in the Disease state to a terminal node and point it to the Dead state.



To the right of each transition node's symbol TreeAge Pro displays the name of the jump-to state for the next cycle.

If the wrong jump-to state has been assigned to a transition node, it is easy to change the specified transition.

► **To change a transition node's jump-to state:**

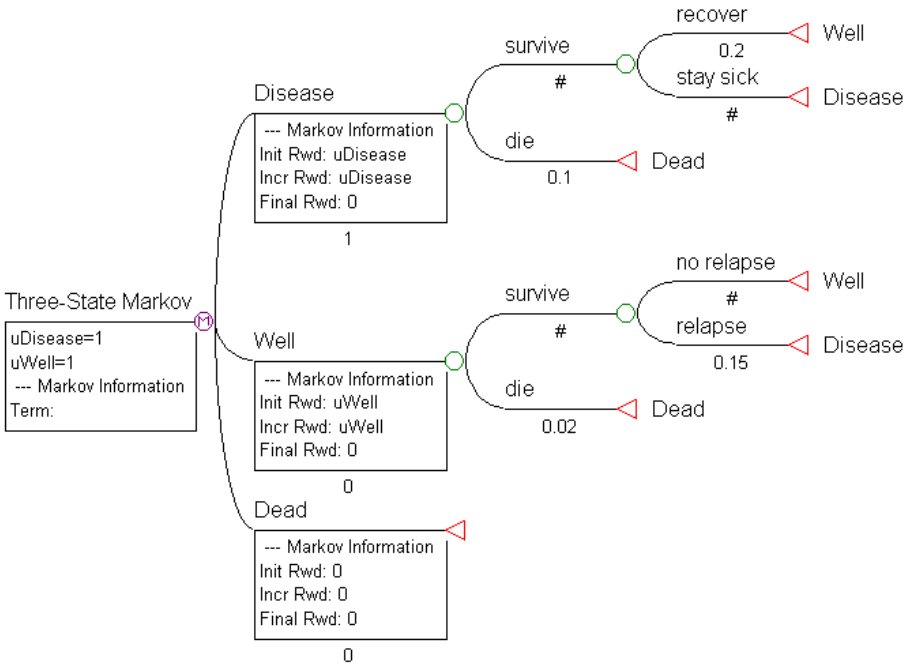
▷ Select an existing transition node in a Markov state transition subtree, and choose **Options >** Markov Jump State…. (Or, double-click on the node symbol.)

Note:
If you change the name of a Markov state, TreeAge Pro automatically updates the transition nodes pointing to it.

Now, set up the Well state's transitions.

- ▷ Change no relapse in the Well state to a terminal node and point it back to Well again.
- ▷ Change relapse to a terminal node and point it to Disease.
- ▷ Change the die node to a terminal node and point it to Dead.

This completes the Markov cycle tree structure.



- ▷ Save the changes to the Three-State Markov model.

The _stage counter and the termination condition

When analyzing a Markov model, TreeAge Pro uses the termination condition, or stopping rule, you specify at the Markov node to determine whether a cohort analysis is complete. TreeAge Pro evaluates the termination condition at the beginning of each cycle except the first. If the condition is *true*, the Markov process ends, final rewards are assigned if necessary, and the results are reported.

The termination condition is usually very simple. Often it just checks how many cycles have been completed, and stops when a certain number is reached. The number of cycles that have passed is contained in a built-in counter called **_stage**, which TreeAge Pro sets equal to 0 before the first cycle and increments by 1 before each subsequent cycle (i.e., before assigning state rewards).

The termination condition can reference variables, and can include multiple conditions, combined using logic symbols: the vertical bar ("|") means OR; the ampersand ("&") means AND; and the exclamation ("!") means NOT. Parentheses can be used to group conditions.

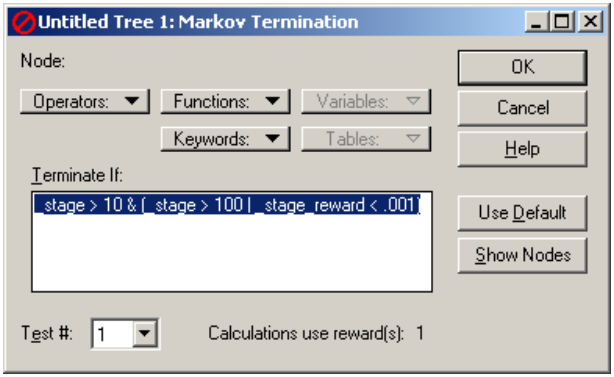
For the Three-State Markov model, use a simple termination condition.

► **To set the termination condition:**

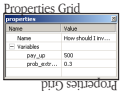
▷ Select the Markov node and choose **Values >** Markov Termination… (or click on the Markov tool bar button, or use the right-click menu).

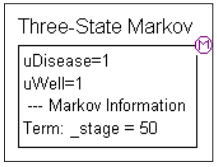


▷ In the Terminate If box, change the termination condition from the default statement to `_stage = 50` and then click OK to return to the tree.



▷ If the properties grid is visible, it can be used to modify the Markov termination condition for the selected node. If necessary, click on the plus sign labeled Markov Information to show the termination condition.





The termination condition _stage = 50 will cause the process to perform 50 cycles (#0 to #49), with an equal number of reward assignments and transitions.

See the next chapter for details on building and interpreting more complex termination conditions.

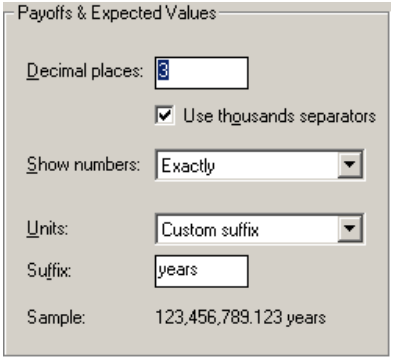
▷ Before continuing, save the changes to the Three-State Markov tree.

Analyzing a Markov model

Once you have completed the Markov model, a number of different kinds of analysis can be performed. Before performing an analysis, however, make sure the tree is set to use appropriate numeric formatting for displaying calculation results.

► **To set the numeric formatting preferences:**

▷ Choose **Edit >** Numeric Formatting…, or press F10.



Set the tree to show at least 3 decimal places for expected values and, optionally, use a custom unit suffix "years"; see Chapter 5 for more details on numeric formatting preferences.

Cohort (expected value) analysis

First, roll back the tree to verify that it is ready to calculate.

▷ Choose **Analysis >** Roll Back.

If you have forgotten to perform one of the steps in the Markov modeling tutorial, an error may be reported identifying the problem node. If there are no errors, TreeAge Pro will display the results of the cohort, expected value calculations on the face of the tree.

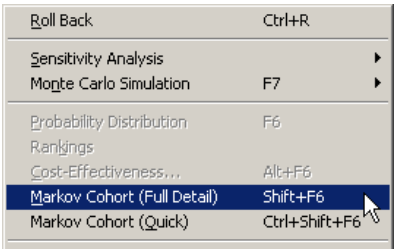
Next to the Markov node a roll back box should display an expected value of about 15.891 years (within the 50 years the model is allowed to run). How this value is calculated will be explained below. The boxes next to the Markov states display the average time spent in the state, along with the final probability (FP) of that state when the process terminates. These numbers correspond to the last row of the cohort analysis text report, explained below.

If roll back is still on, turn it off before continuing with another analysis.

▷ To turn off roll back, choose **Analysis >** Roll Back.

Usually the preferred expected value analysis for a Markov model is a Markov cohort analysis at the Markov node. It provides detailed text report options, and a variety of graphical outputs.

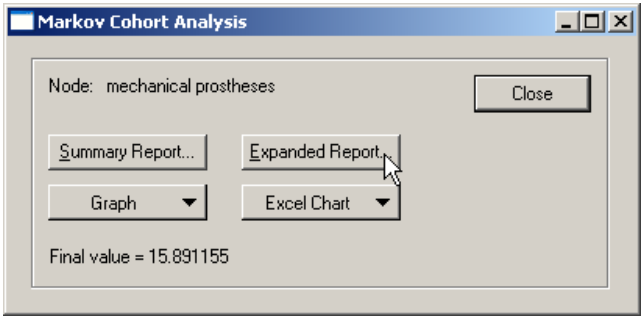
► **To perform a cohort expected value analysis:**



▷ Select the Markov node.

▷ Choose **Analysis >** Markov Cohort Analysis. You can choose between the "Full Detail" and "Quick" forms of analysis. The difference between them is described below.

After the cohort analysis terminates, the Markov Cohort Analysis output window is displayed, showing the expected values calculated at the Markov node, and offering several output options. Starting with TreeAge Pro 2005 if you run the "Full Detail" cohort analysis, an additional option is available: the Expanded Report.

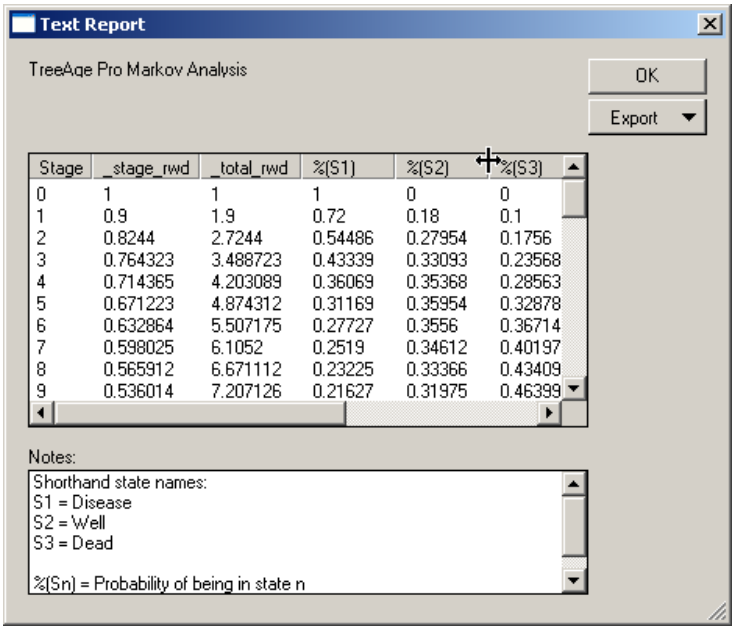


Charting with Microsoft® Excel:

If you have the TreeAge Pro Excel module, TreeAge Pro graphs can instead be created as Excel charts and text reports can be exported to a spreadsheet, all with a single click. In some cases, Excel charts have additional functionality. See the Excel module's documentation for details.

Here is a summary of the Markov Cohort Analysis outputs:

- **Summary (or Text) Report** – The summary Markov text report includes per-state probabilities and rewards. Its contents can be exported for printing or further analysis using a spreadsheet or statistical software package. The last row reflects the expected values and final state probabilities. See the graphic on the next page.

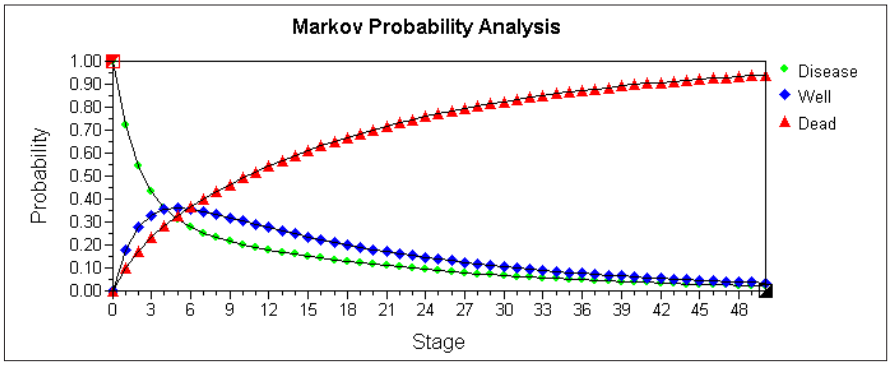


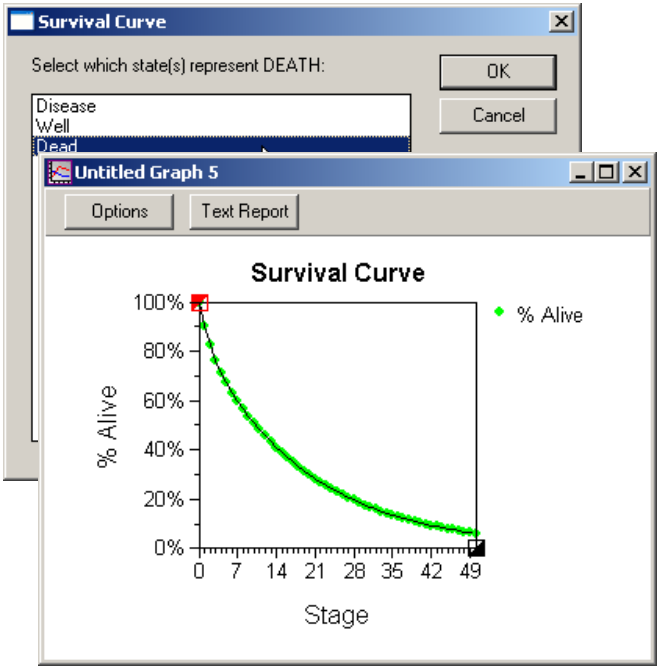
- **Expanded Report** – A much more detailed version of the Markov text report. The Expanded Report goes beyond the standard, cycle-by-cycle summary report by specifying the calculated transitions for every event in every state. It also provides additional per-cycle reward information in cost-effectiveness and multi-attribute Markov models. A detailed explanation of the Expanded Report and its options is provided later in this chapter.

Tip:

The "Quick" form of Markov Cohort Analysis will save calculation time in those situations where the expanded report will not be required.

- **State probabilities graph** – This graph plots the changing state probabilities at each cycle, and is closely related to the survival curve.





- **Survival curve** – Survival curves are a standard means of communicating the results of a Markov analysis. TreeAge Pro will prompt you to select which states represent death, and then will plot the sum of the "alive" state probabilities. This graph can also be used to create probability curves that group other kinds of states — to plot disease-free survival, for example.
- **State rewards graph** – This graph shows, for each state, what reward was received at each stage. (For cost-effectiveness models, cost and effectiveness are plotted separately.)
- **_stage_reward and _total_reward graphs** – Each graph contains a single line plotting the value of the specified keyword at each cycle.

Expanded Report

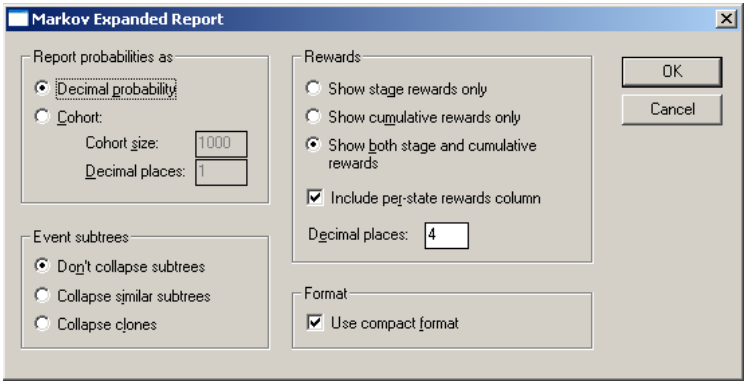
Various format options can be specified for the expanded report.

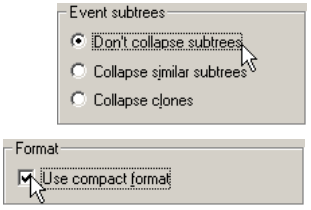
The "Report probabilities as" settings determine whether events are displayed using decimal probabilities or as numbers representing probabilities multiplied by an arbitrary cohort size. The "Rewards" options determine which columns of per-cycle and cumulative reward information are reported to the right of the transition information. The "Use compact format" option can be used to eliminate some white space in the report.

The "Event subtrees" options can be used to summarize probability information. The display of events and their probabilities in the expanded report can be simplified in many models by specifying that similar or cloned events be aggregated ("collapsed"), both within a single state and even across states.

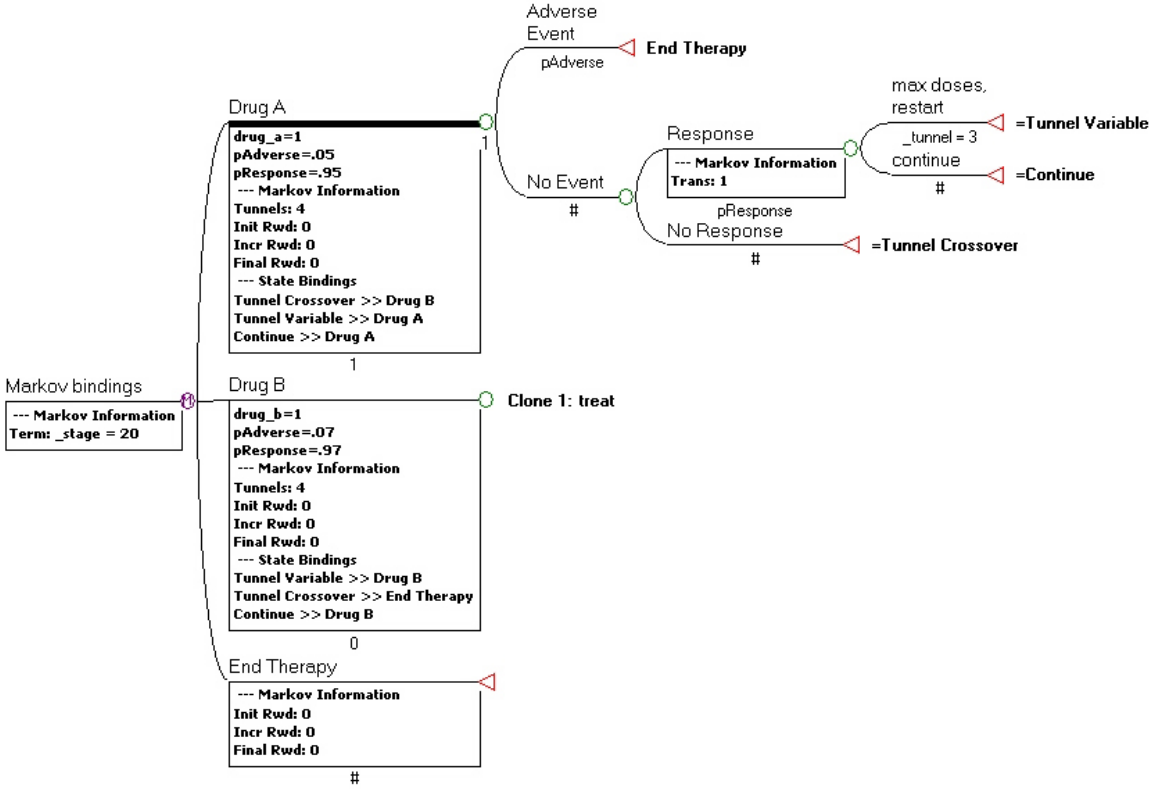
A Markov model is shown on the following page, along with sections from two expanded reports generated from a Markov cohort analysis. (The model's structure is described in the section on tunnel states in Chapter 35.) These two example reports, along with the two more examples shown on the next page, illustrate how the various Event subtree options can be used to customize the layout of the expanded report.

The default Event subtree option, "Don't collapse…", results in every event in every state being represented separately in the expanded report. Both examples on the following page use this default setting; the difference between the two uncollapsed reports shown is that the bottom report illustrates the slight compression offered by the Use compact format setting. This setting does not affect the content of the report, but merely tightens formatting by eliminating some white space.



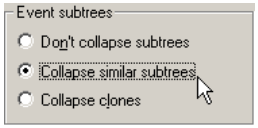


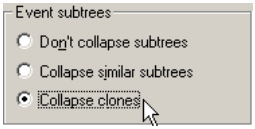
Tip:
If you use a weighted multi-attribute calculation method, possibly in a cost-effectiveness model, use the Expanded Report with full reward detail to see the per-cycle rewards broken down by attribute contributions.



Stage	Start state	Start prob...									End state	End p
0	Drug A	1	[Drug A]	1	[Adverse Event]	0.05					Drug A	0.902
			Adverse Event	0.05	* End Therapy	0.05					Drug B	0.047
			No Event	0.95							End Therapy	0.05
					[No Event]	0.95	[Response]	0.9025	[max doses, restart]	0		
					Response	0.9025	max doses, restart	0	* Drug A	0		
					No Response	0.0475	continue	0.9025				
									[continue]	0.9025		
							[No Response]	0.0475	* Drug A	0.9025		
							* Drug B	0.0475				
1	Drug A	0.9025	[Drug A]	0.9025	[Adverse Event]	0.04512					Drug A	0.814
	Drug B	0.0475	Adverse Event	0.04512	* End Therapy	0.04512					Drug B	0.085
	End The	0.05	No Event	0.85737							End Therapy	0.099

Stage	Start state	Start prob...									End state	En
0	Drug A	1	[Drug A]	1	[Adverse Event]	0.05	[Response]	0.9025	[max doses, restart]	0	Drug A	0.9
			Adverse Event	0.05	* End Therapy	0.05	max doses, restart	0	* Drug A	0	Drug B	0.0
			No Event	0.95			continue	0.9025			End Therapy	0.0
					[No Event]	0.95			[continue]	0.9025		
					Response	0.9025	[No Response]	0.0475	* Drug A	0.9025		
					No Response	0.0475	* Drug B	0.0475				
1	Drug A	0.9025	[Drug A]	0.9025	[Adverse Event]	0.04512	[Response]	0.81451	[max doses, restart]	0	Drug A	0.8
	Drug B	0.0475	Adverse Event	0.04512	* End Therapy	0.04512	max doses, restart	0	* Drug A	0	Drug B	0.0
	End Therapy	0.05	No Event	0.85737			continue	0.81451			End Therapy	0.0





If a model includes similar events found at the same generation in one or more states' transition subtrees, using the Collapse similar… option will, to the extent feasible, merge these events and their probabilities in the text report. TreeAge Pro determines similarity based on the number of branches and the branch descriptions/labels. In order for events to be merged, their entire subtrees must be similar.

In a Markov process using clones, using the report's Collapse clones option will work similarly to the Collapse similar… setting, but merging only cloned transition subtrees. The graphics below illustrate the effect of the Collapse clones option, with the cloned transition subtrees having been merged in the bottom version.

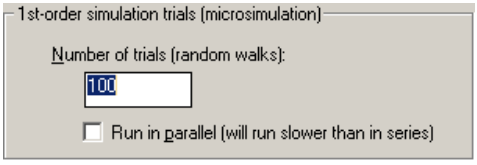
If you generate the report and find that you want to change a display option, simply close the text report dialog to return to the analysis output window, and click Expanded Report... again.

Stage	Start state	Start prob...								
1	Drug A	0.9025	[Drug A]	0.9025	[Adverse Event]	0.04512				
	Drug B	0.0475	Adverse Event	0.04512	* End Therapy	0.04512				
	End Therapy	0.05	No Event	0.85737						
					[No Event]	0.85737	[Response]	0.81451	[max doses, restart]	0
					Response	0.81451	max doses, restart	0	* Drug A	0
					No Response	0.04287	continue	0.81451		
									[continue]	0.81451
			[Drug B]	0.0475	[Adverse Event]	0.00333	[No Response]	0.04287	* Drug A	0.81451
			Adverse Event	0.00333	* End Therapy	0.00333	* Drug B	0.04287		
			No Event	0.04418						
					[No Event]	0.04418	[Response]	0.04285	[max doses, restart]	0
			[End Therapy]	0.05	Response	0.04285	max doses, restart	0	* Drug B	0
			* End Therapy	0.05	No Response	0.00133	continue	0.04285		
									[continue]	0.04285
							[No Response]	0.00133	* Drug B	0.04285
							* End Therapy	0.00133		

Stage	Start state	Start prob...								
1	Drug A	0.9025	[Drug A]	0.9025	[Adverse Event]	0.04845				
	Drug B	0.0475	Adverse Event	0.04512	* End Therapy	0.05178				
	End Therapy	0.05	No Event	0.85737						
					[No Event]	0.90155	[Response]	0.85736	[max doses, restart]	0
					Response	0.85736	max doses, restart	0	* Drug A	0.81451
					No Response	0.04419	continue	0.85736		
									[continue]	0.85736
			[Drug B]	0.0475			[No Response]	0.04419	* Drug A	0.85736
			Adverse Event	0.00333			* Drug B	0.04419		
			No Event	0.04418						
			[End Therapy]	0.05						
			* End Therapy	0.05						

A note on microsimulation

Roll back and cohort analysis both evaluate Markov models using cohort calculation methods. Another way to evaluate a Markov model is using Monte Carlo microsimulation. Microsimulations (first-order trials) use random number sequences to send one individual at a time on single paths through the model. This approach is required used by some complex Markov models, in order to keep track of an individual's history (beyond what state they are in).


1st-order simulation trials (microsimulation)
Number of trials (random walks):
100
Run in parallel (will run slower than in series)

Because individuals are randomized based on probabilities, each simulation of a model returns a different set of results, but the summary statistics should converge if a sufficient number of trials are run. The more complex a model is (i.e., more states, more cycles), the more trials that will be required to approach the mean reported by cohort analysis.

Mean Value:	16.930 years
Std Dev:	15.494 years
Minimum Value:	1.000 years
2.5%:	1.000 years
Median:	11.000 years
97.5%:	50.000 years
Maximum Value:	50.000 years

The simulation results for the Three-State Markov model will report only integer life expectancies for the simulated individuals/trials, as you would expect with a model that only assigns integer rewards.

See Chapter 16 for general information on Monte Carlo simulation; see Chapter 36 for more information on specific issues related to Markov Monte Carlo microsimulation.

A note on half-cycle correction

An important assumption made in discrete-time Markov models is that all state transitions occur simultaneously, at the end of each cycle. In reality, however, most kinds of transitions typically occur gradually throughout a time interval (on average, half-way through). This assumption does not affect reported probabilities (or the survival curve), but it may result in overestimation of expected survival in most models.

The error will be greater in cases where there is a significant difference between the incremental reward associated with an individual's starting state and the reward associated with the jump-to state, as there is between alive and dead states. Assuming that transitions occur halfway through a cycle on average, the technically ideal correction is to assign a half-reward corresponding to the current state, and then a half-reward at every transition node (transition rewards are described in the next chapter) corresponding to the cost or utility associated with the *jump-to* state.

Most models, however, use a simpler adjustment, called the *half-cycle correction*. To implement the half-cycle correction, you can simply divide every alive state's incremental reward in half, and assign the half-reward as its initial and final reward.

In the Three-State Markov example, this will result in a correction of about one-half of a cycle's reward, or about 0.5 years; try the Markov cohort analysis after making this change.

See the next chapter for more on how to use the half-cycle correction.

Cost-effectiveness Markov models

Building a cost-effectiveness (CE) Markov model is quite similar to working with a regular CE decision tree (see Chapters 31 and 32). First, you must set the appropriate preference settings, including calculation method and numeric formatting.

► **To set up a Markov model for cost-effectiveness calculations:**

▷ Choose **Edit >** Preferences…, and set the Calculation Method to Cost-Effectiveness, noting which payoff/reward set is used for cost and which for effectiveness.

▷ Click the **Numeric Formatting…** button to open the Numeric Formatting dialog and set the appropriate formatting for costs, effectiveness, and ICERs.

Separate sets of rewards must be entered in the Markov model for the cost and effectiveness attributes. Just as each terminal node in a decision tree can use up to nine payoff expressions, each Markov subtree has nine corresponding reward sets. In CE calculations, reward set #1 might be used for costs and reward set #2 for effectiveness, but this is flexible. If you already have a single-attribute Markov model using payoff set #1 for effectiveness, simply set CE calculations to use payoff set #2 for costs. As described in Chapter 32, you also have the option of specifying in the CE preferences that multiple payoffs be combined for cost.

At each Markov state, the three reward types described earlier — initial, incremental and final — can be entered for each reward set.

► **To assign cost and effectiveness state rewards:**

▷ Select a Markov state and choose **Values >** Markov State Information….

▷ In the Markov State Information dialog, assign values or expressions for both cost and effectiveness state rewards.

Be sure to check the accuracy of the information specified below the three state reward text editors, in the pop-up menu labeled Rewards. Use this menu to select which of the reward sets to edit. To switch between the cost reward set and the effectiveness reward set, simply select the appropriate reward set number from the pop-up menu.

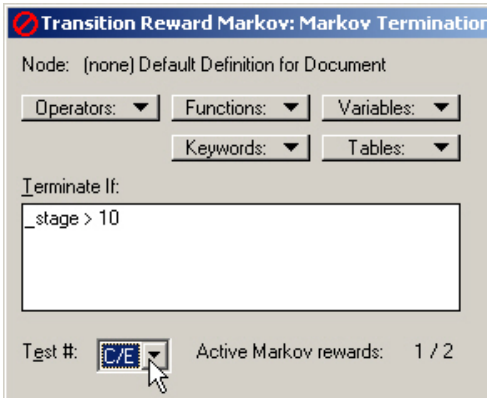
Next to the pop-up menu, TreeAge Pro specifies which reward sets are active in tree calculations – for example, "Active Markov rewards: 1 / 2" indicates that reward set #1 will be used for calculating costs, and reward set #2 will be used for calculating effectiveness values. Changing the Rewards pop-up menu does *not* change the active calculation method settings, which can only be changed from the Preferences dialog, as described above.

Additionally, a CE Markov model requires that you enter a distinct termination condition for the Cost-Effectiveness calculation method. TreeAge Pro maintains separate termination conditions for each Simple, single-attribute calculation, and for Cost-Effectiveness.

► **To assign a cost-effectiveness termination condition:**

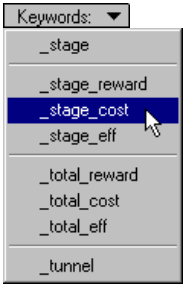
▷ Select the Markov node, and choose **Values >** Markov Termination.

▷ Enter the termination condition in the dialog box, and click OK to close the dialog.



Cost-effectiveness keywords

There are several Markov keywords available only in a cost-effectiveness model. The keywords ***_stage_cost***, ***_stage_eff***, ***_total_cost***, and ***_total_eff*** calculate the single-attribute values; ***_stage_reward*** and ***_total_reward*** calculate CE ratios, and not often used in CE Markov models.



The default termination condition which TreeAge Pro automatically enters the first time you open the Markov Termination dialog at a Markov node under the cost-effectiveness calculation method, is _stage > 10 & (_stage > 100 | _stage_eff < 0.001). The **_stage_eff** keyword is used in the default condition instead of **_stage_reward**, based on the fact that most common effectiveness calculation is life expectancy (perhaps quality-adjusted), and that a useful limiting point for the process in such cases is when approximately 99.9% of the cohort is dead.

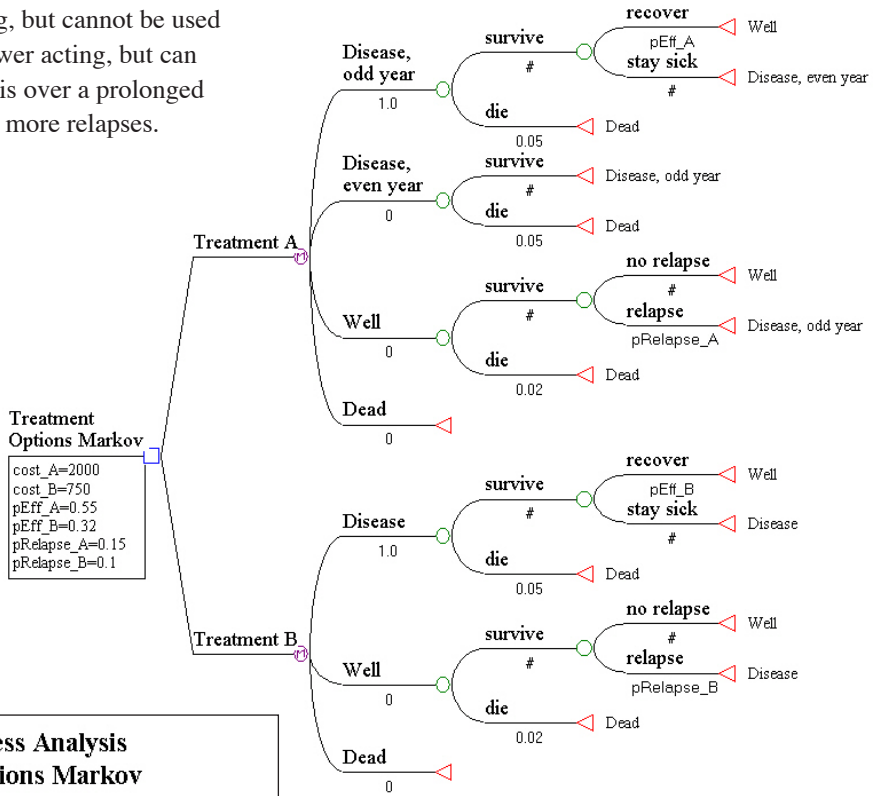
> ***Tip:***
> Always change the termination condition to something appropriate for your model! In many models, no effectiveness threshold is needed, and the termination condition will only reference the _stage keyword. If your effectiveness measure is a rare event counter, then either no effectiveness threshold or a lower one might be required.

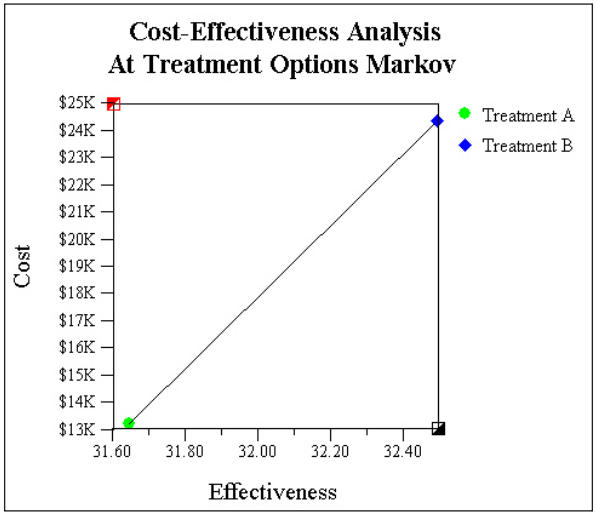
A cost-effectiveness Markov model

The example tree entitled Treatment Options Markov, found in the Tutorial Examples\Healthcare subdirectory, compares the cost-effectiveness of two hypothetical treatments, using a similar disease model to that of the Three-State Markov process. The tree is shown below right.

In this still relatively simple model, Treatment A is presumed to be faster acting, but cannot be used long term. Treatment B is slower acting, but can be used on a maintenance basis over a prolonged period, effectively preventing more relapses.

You can select one of the Markov nodes at a time and perform a Markov cohort analysis, or select the root, decision node and perform a cost-effectiveness analysis (see Chapter 31), which will yield a graph like that shown below the tree.





Chapter 35

Markov Modeling Tools and Techniques

This chapter covers a number of commonly-used features in Markov models, including using tables of probabilities and discounting rewards. It also covers a number of Markov modeling features that are infrequently used, but may be indispensable in some cases.

Markov Monte Carlo microsimulation trials are covered in Chapter 36, and miscellaneous Markov topics are covered in Chapter 37.

In this chapter:

- Keywords, time-dependence, and discounting 470
- Probability functions 474
- Assigning onetime costs and utilities 476
- Cloning Markov models 480
- Counting "time in state" with tunnels 484
- Markov decision processes 491
- Advanced Markov modeling and analysis options 494

Keywords, time-dependence, and discounting

TreeAge Pro provides several Markov *keywords* — built-in variables which are available only in a Markov node or its subtree. The first two listed are integer counters:

- **_stage** – the number of the cycles that have passed (starts at 0 for first cycle)
- **_tunnel** – the number of cycles spent continuously in a *tunnel state*
- **_stage_reward** – the reward received by the cohort in the previous cycle (in Simple calculations)
- **_stage_cost, _stage_eff** – counterparts of _stage_reward in Cost-Effectiveness (CE) calculations
- **_total_reward** – the cumulative reward of all previous cycles; at the end of calculations, this is the overall value of the Markov process
- **_total_cost, _total_eff** – counterparts of _total_reward in CE calculations
- TreeAge Pro also includes a special function, **StateProb( )**, for accessing current state probabilities during analysis.

As illustrated in the previous chapter in the tutorial on building the Three-State Markov model, the _stage counter is useful in defining the Markov termination condition. This section will describe other important functions of the _stage counter and other keywords.

See Chapter 37 for a detailed look at the Markov process algorithm in TreeAge Pro, indicating when and in what order Markov keyword values are modified, both during cohort analysis and Monte Carlo microsimulation (first-order trials).

Cycle zero

In TreeAge Pro Markov models, the first cycle is referred to as cycle 0 and the _stage counter is equal to 0 during this first cycle. For example, if a model's cycle length is one year, cycle 0 represents the first year of the process; if this process started with an individual's birth, cycle 0 would correspond to an age of 0 – i.e., the year prior to an individual's first birthday.

The following events occur during the first cycle of a Markov process, while the keyword _stage is equal to 0:

- the cohort is distributed among the Markov states according to the initial probabilities entered under the branches (the *only* time these probabilities are used);
- initial rewards are accumulated based on state membership;
- the members of a state traverse the transition subtree based on the transition probabilities, and the percentage of the cohort at a transition node are assigned the transition rewards in the path back to the state (before entering new states for the next cycle);

You should ensure that references to tables in initial and transition probability expressions, as well as in initial state rewards and transition rewards, will work correctly when _stage = 0.

The *incremental* state reward expressions are *not* accumulated during cycle 0; only the *initial* rewards are evaluated. The initial probabilities determine which states are populated in cycle 0, and therefore where initial state rewards are required. If half-cycle correction is not used, the initial state reward for a state is often the same as the incremental reward; see the section on half-cycle correction later in the chapter for more details.

Using tables of time-dependent transition probabilities – an example

The Three-State Markov model from the previous chapter is an example of a *Markov chain* — a Markov model in which all probabilities and other parameters remain constant over time. In the kinds of Markov models used to represent healthcare issues, however, probabilities and other values often vary over time. This kind of model is referred to sometimes as a Markov process.

In the TreeAge Pro Healthcare module, any expression in a Markov model (not just the termination condition) can reference tables of stage-dependent values using the _stage counter. Other kinds of time-dependent expressions can also be created, using the _tunnel counter, tracker variables, etc.

This tutorial requires two things:

- A copy of the Three-State Markov model you created in the previous chapter. If you did not build the tree, you can find a copy of it in TreeAge Pro's Tutorial Examples\Healthcare_backup subdirectory.
- A table file called "tMort" to hold time-varying probabilities. Follow the instructions below to create the table. If you have additional questions about working with tables, refer to Chapter 18.

Important:

The table is created, initially, as a file stored separately from the tree, in TreeAge Pro's Tables subdirectory. You are urged to review Chapter 18 to learn more about working with tables. Chapter 18 describes how to package a tree and its required tables into a single file, for example.

► **To create a new table for use in a tree:**

▷ In the tree window, choose **Values >** Variables and Tables…. In the Variables and Tables dialog, make sure the Tables List option is selected.

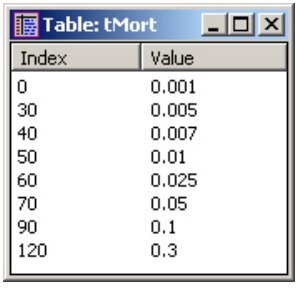
▷ Click **New Table…**, and enter `tMort` for the new table's name. Use the same file name.

▷ Click OK to create the empty table file and return to the Variables and Tables list.

▷ With tMort selected in the list, click the **Open Table…** menu button and choose Edit Table…. This will open the Table editing window.

▷ In the Table window, use the Add Table Entry dialog to enter the index/value pairs shown at left. You may enter multiple values in quick succession by using the **More** button. (See Chapter 17 for instructions on pasting tab-delimited text into the Table window.)

When you click OK, the pairs you entered in the Add Entry dialog will be displayed in the table window. If you need to enter more pairs in the table, or make other changes while in the Table window, use the **Actions** or **Table** menu commands.

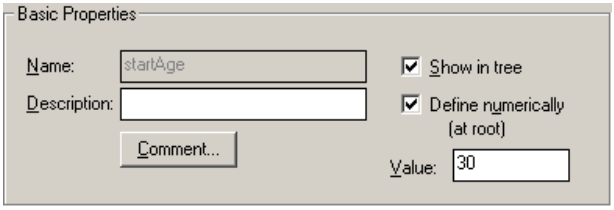


Now, update the Three-State Markov model to use the new table of mortality probabilities.

► **To look up a transition probability in a table:**

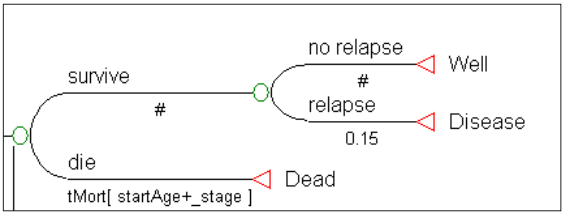
▷ Select the die branch of the Well state, and change its probability to the formula `tMort[startAge+_stage]`.

▷ Press ENTER or TAB to have TreeAge Pro check the new probability expression. TreeAge Pro will prompt you to create the variable startAge. In the Basic Properties for startAge, specify a default numeric value of 30, and click OK.



Note that, had you not already created the table tMort, TreeAge Pro would also ask if you want to create it. In this case you would follow the steps in Chapter 18 showing how to create and populate a table.

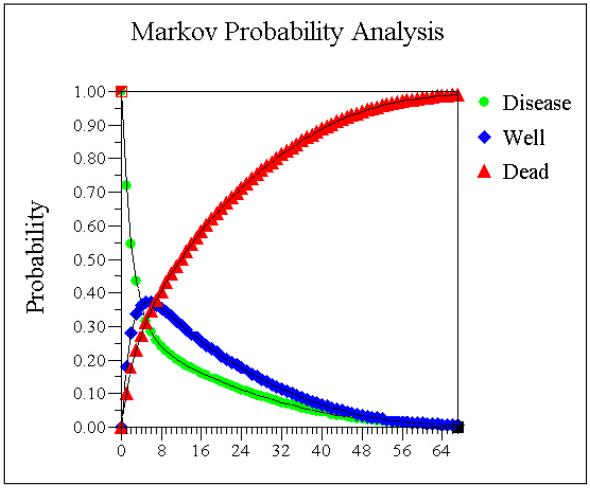
Now, in place of a fixed probability of death from other causes, TreeAge Pro will calculate the transition probability at every cycle using the table lookup tMort[startAge+_stage]. The first set of transitions in the Markov process, when _stage = 0, will use the value returned by the reference tMort[30+0], which is 0.005.



Each subsequent cycle will use a higher mortality probability, because the values in the tMort table increase as the indexes increase. Also relevant is this case is the fact that the table is currently set to interpolate between existing indexes, when you reference a missing index. In the example, the missing value for cycle 1 when the table reference is tMort[31] will be calculated using linear interpolation between the table values for indexes 30 and 40. The interpolated probability will be 0.0052. Missing rows at subsequent cycles will be similarly calculated.

If you now roll back the tree, the Markov node should display an expected value of 17.077 – somewhat higher than the roll back value calculated in the previous chapter, when the Three-State Markov process was still based on an unchanging probability of mortality.

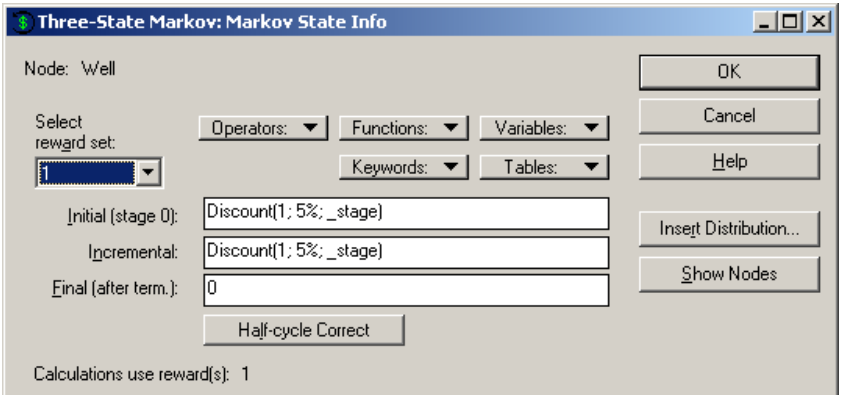
If you run a Markov cohort analysis in the new version of the tree, and compare the new state probabilities graph with the graph generated in the previous chapter, you will see that the graphs have a similar shape, but that the new process is terminating earlier. Despite its higher expected value, the new Markov process drops below the stage reward threshold around stage 65, rather than after stage 100.



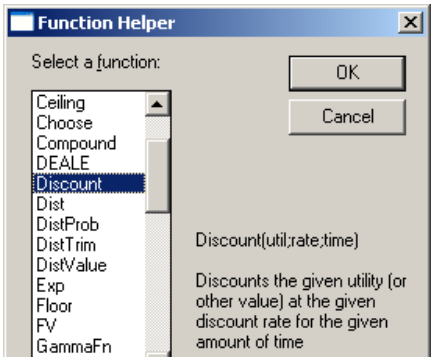
Discounting rewards

In addition to defining stage-dependent probabilities, tables can also describe stage-dependent rewards, such as costs or utilities. However, if all you need to do is *discount* costs and utilities, a simple exponential formula may be used instead of a table. For example, the expression costX /((1+rate)^_stage) can be used to discount the reward value at each stage. If the cycle length is not equal to the period of the discount rate (usually 1 year), then _stage should either be divided (for shorter cycle lengths) or multiplied (for longer cycle lengths).

However, rather than entering an exponential formula like the one shown above, you can use TreeAge Pro's built-in discounting function, Discount(). This function takes three parameters: *value*, *rate*, and *time*. For example, the expression Discount(costX; rate; _stage) will yield the same result as the exponential formula shown above. The Discount() function is equally applicable to the discounting of costs and utilities.



TreeAge Pro's built-in functions can be inserted into formulas either by typing them in yourself, or by using the Functions button found in most expression editors. TreeAge Pro provides assistance in correctly placing the required parameters in a function if you choose Functions > Helper….



See Chapters 13 and 14 for details on using the Functions Helper with TreeAge Pro's built-in functions.

Probability/rate functions

The following functions are used to convert between rates and probabilities (or odds).

They generally employ one or both of the arguments *rate* (or prob) and *time*. In each case, it is essential that the values for these parameters be based on the same scale. For example, if a rate being converted is in terms of years (such as yearly mortality), the time parameter must also be in years.

Function	**Explanation**
DEALE(rate; time)	DEALE is an acronym for "declining exponential approximation of life expectancy."
	Formula: $(1 - e^{-rate*time}) / rate$
OddsToProb(odds)	Converts odds into a probability.
	Formula: odds / (1 + odds)
ProbFactor(prob; fac)	First converts the probability to odds, then multiplies it by the given factor, then converts it back to a probability. See OddsToProb and ProbToOdds.
ProbToOdds(prob)	Converts a probability into odds.
	Formula: prob/ (1 - prob)
ProbToProb(prob; multiplier)	Converts a probability into a rate, multiplies the rate by the given multiplier, and converts back to a probability.
	*Equivalent to: RateToProb((ProbToRate(prob; 1) * multiplier); 1)*
ProbToRate(prob; time)	Converts a probability into a rate, and divides the rate by time.
	Formula: - ln(1 - prob) / time
RateToProb(rate; time)	Multiplies a rate by time, and converts it into a probability.
	Formula: $1 - e^{-rate*time}$

Two additional functions with "Prob" in their names have special application in Markov models. See the section on Dynamic cohort models later in this chapter for details.

PathProb()	Returns the cumulative path probability for the node being calculated (the Markov node in a Markov subtree).
StateProb(A)	Returns the state probability of state #A at the start of the current cycle. See the end of this chapter for details.
StateProb(A; B)	Returns the state probability of the set of states from #A o #B. See the end of this chapter for details.

Notes on probability/rate functions

These functions are sensitive to user errors. You are urged to exercise great care when using them in your models. It is recommended that you use the Calculator/Evaluator or Variables Report features, located under TreeAge Pro's Values menu, to test expressions using these functions; see Chapter 13.

RateToProb(), as the name suggests, is used to convert a rate into a probability, either for the same time period (time=1) or a different time period. For example, if a disease being modeled has a yearly mortality rate of .05, you could convert this to a probability using the formula:

RateToProb(.05; 1)

ProbToRate() provides the reverse function to RateToProb(), ultimately enabling a probability to be converted to a rate, proportionally increased or decreased using the additive property of rates, and then converted back to a probability.

In both RateToProb() and ProbToRate(), the second, time parameter will allow you to convert between a rate for an interval of one length and a probability for a interval of a different length.

The DEALE() function is cumulative, so the time parameter means "over the course of this amount of time."

Assigning one-time costs and utilities

There are a number of different situations which may require assigning a *onetime* reward in a Markov model, rather than an incremental reward for each cycle spent in a particular state.

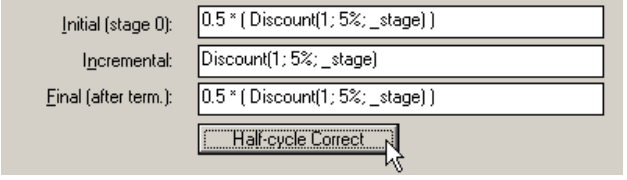
Half-cycle correction

Real processes occur in continuous time, with transitions and other events occurring throughout an interval of time. In TreeAge Pro, however, a Markov process occurs as a discrete sequence of snapshots, with transitions understood to occur at the end of each cycle. As described in the note on half-cycle correction in the previous chapter, in an absorbing Markov process where everyone dies eventually, an uncorrected expected value calculation will overestimate life expectancy by about half of a cycle (0.5 years in a one-year cycle length model). The explanation for this is relatively simple. *In whatever cycle a "member" of the cohort analysis dies, they have already received a full cycle's worth of state reward, at the beginning of the cycle*. In reality, however, deaths will occur halfway through a cycle on average. So, someone that dies during a cycle should lose half of the reward they received at the beginning of the cycle (e.g., -0.5 years of life expectancy in a one-year cycle length model).

Instead of implementing the half-cycle correction as a toll at each transition to death, however, it is easier to implement it in an absorbing process simply by subtracting a half-reward from the rewards assigned at the beginning of the process, in cycle 0 — i.e., by setting a state's initial reward to one-half of its incremental reward. This is the primary, though not only, reason that the state rewards are separated into three parts.

In a non-absorbing process, in which a significant percentage of the cohort may be alive when the process terminates, cohort members still alive at the end of the process should be given back the half-cycle "death" correction taken from their initial reward at the beginning of the process. This is done by adding on a half-reward after termination, in the final reward for all alive states (it does not hurt to always include the initial and final components at every state).

To have TreeAge Pro multiply the Incremental reward expression by half and put it in the initial and final rewards, simply click the **Half-cycle Correct** button.



Any reward that is a function of life expectancy (i.e., medication costs that occur gradually over a cycle) is usually corrected in the same way.

Note:

In models that calculate quantities other than simple life expectancy, for example *quality-adjusted* life expectancy, different alive states will have different rewards. This means that a perfect half-cycle correction might require correcting not just for death transitions, but for other kinds of transitions from higher value states to lower value states (i.e., where someone should receive half a cycle of the starting state's reward and half of the ending state's reward). Note, however, that Markov approximation errors in two strategies will often cancel each other out in incremental calculations, and reasonable judgement should be used to decide when to half-cycle correction.

Prior costs

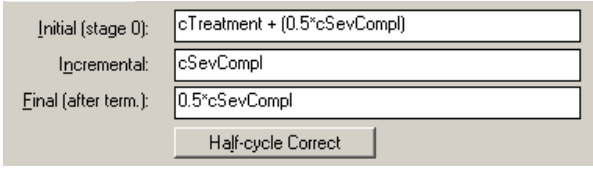
In some models, it is necessary to account for costs, utilities, or life expectancy that occurred prior to the Markov process. Consider, for example, a tree which deals with the uncertainties associated with a particular treatment. In this model, a Markov process will be encountered only if a particular event occurs. In the standard tree structure, costs are incorporated into a payoff formula at terminal nodes. In the scenario including the Markov model, though, these costs must also be accounted for in Markov rewards.

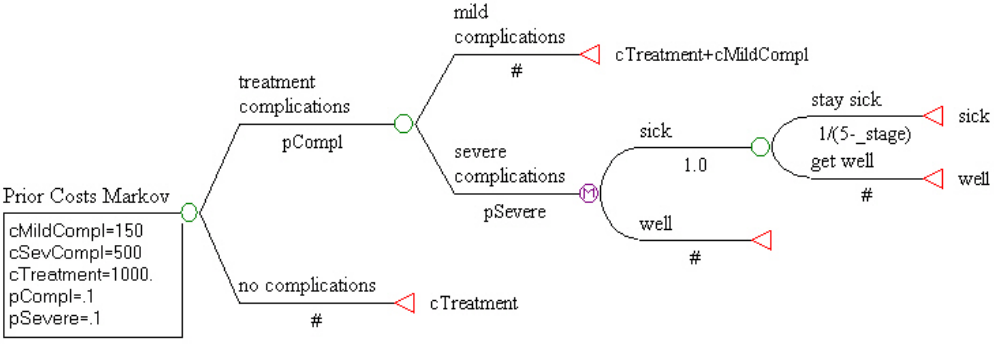
Typically, prior value expressions should be entered in the initial state reward of all states with a non-zero initial probability. This would ensure, for example, that all members of the cohort receive the prior costs. In a cost-effectiveness model (or any model with multiple attributes) be sure to use the appropriate reward set. Cost-effectiveness Markov models are discussed later in this chapter. If your model also uses the half-cycle correction (see above), the initial reward expressions must combine the prior values and the half reward.

► **To include prior costs in a Markov model:**

▷ Create and define a variable or expression that represents all costs accumulated before the Markov process.

▷ For each state with a nonzero initial probability, update the initial reward expression to add the prior costs expression, remembering to keep the half-cycle correction if needed.





Transition rewards

In some models, you may need to account for a cost or disutility associated with a transient event, rather than a state. In many such cases, a transition reward can be used. Transition rewards can be assigned at any node to the right of the Markov state nodes (not just the actual transition nodes).

For instance, a onetime cost may be associated with admission as an inpatient. This cost is not incremental, and should not be accumulated in each interval spent in the hospital. Nor can the cost be assigned using an initial state reward if the admission event is not just an initial, cycle 0 event (initial rewards are only assigned when _stage = 0).

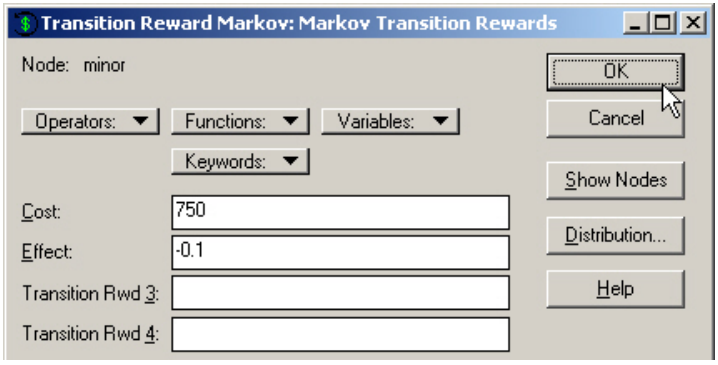
Another example might be a relatively minor complication event during treatment. Although the complication is not a state itself, and may have no effect on state transition, it may have costs and/or disutilities associated with it.

In the example Transition Reward Markov tree, transition rewards are specified both for costs (in reward set #1) and for effectiveness (in reward set #2).

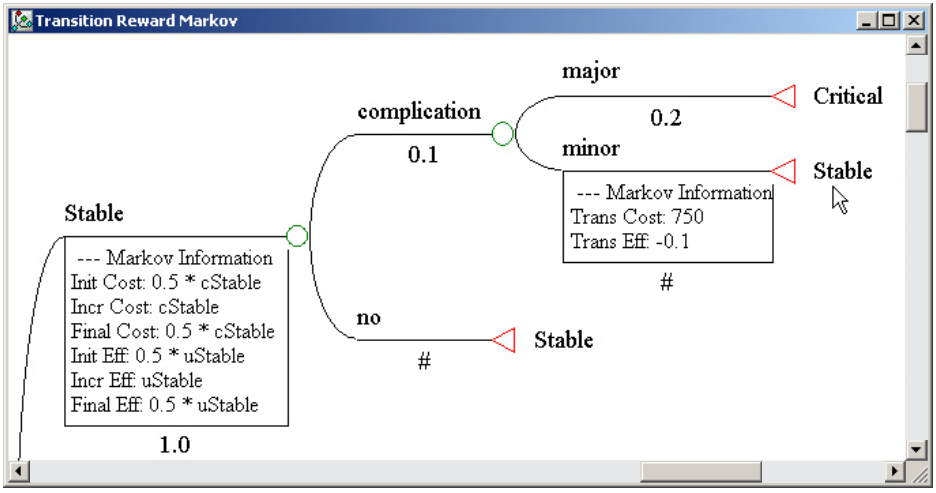
► **To assign a transition reward:**

▷ Select the node where the event occurs, to the right of a Markov state, and choose **Values >** Markov Transition Rewards… (or right-click on the node).

▷ Select the appropriate reward set and enter a reward expression. Leave unused transition rewards empty, rather than entering 0. Press ENTER or click OK.



If the Markov information display preference is turned on, transition reward expressions are shown below the branch.



Notes on transition rewards:

- Transition rewards are not accounted for separately from state rewards.
- Transition rewards are associated with the *cycle* in which they occur.
- In cohort analysis reports, transition rewards are not associated with the *state* in which they occur; instead, they are probabilistically divided among the Markov states to which the transition may lead.
- Like state rewards, transition rewards are *added* to the net reward. Thus, transition rewards should be entered using the appropriate sign, positive or negative. For example, transition costs are normally entered as positive numbers, while transition disutilities are normally negative numbers.

Cloning Markov models

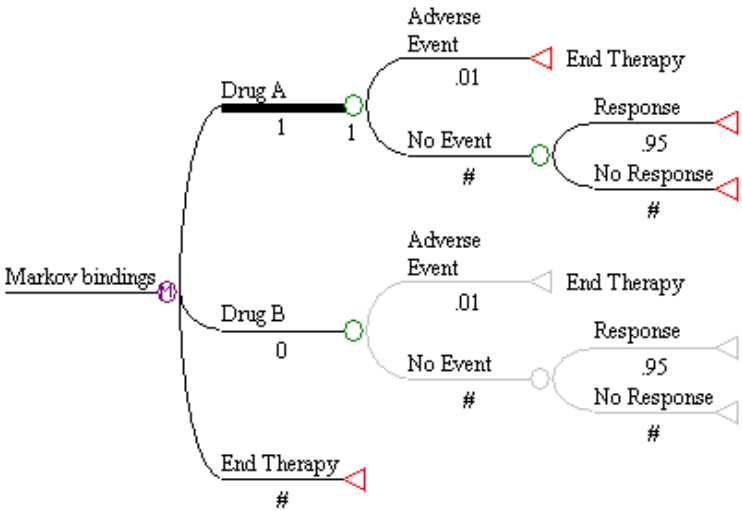
A major advantage of using clones in any tree is the ability to reuse a particular structure in multiple parts of a tree, retaining the option to vary probability and other value expressions in each "copy," but only having to maintain the master subtree (see Chapter 21). In TreeAge Pro, entire Markov models can be cloned within a single tree, and parts within a complex transition subtrees can also be cloned.

Using Markov transition bindings

Clone copies of Markov transition subtrees will, by default, employ the jump-to state settings specified in the clone master. In TreeAge Pro, Markov bindings can be used to have a transition in a clone copy use a different jump-to state that the clone master.

As described in Chapter 21, variables can be used in clone copies when numeric values such as probabilities should not be controlled by the clone master. Markov bindings function similarly; rather than assigning a numeric value to the binding name, a Markov state name is assigned instead. Like variable definitions, Markov bindings must be defined at an appropriate node. Markov bindings may be defined at any node on the Markov subtree, including at the Markov node. Typically, they are created at the root nodes of the clone master and clone copies.

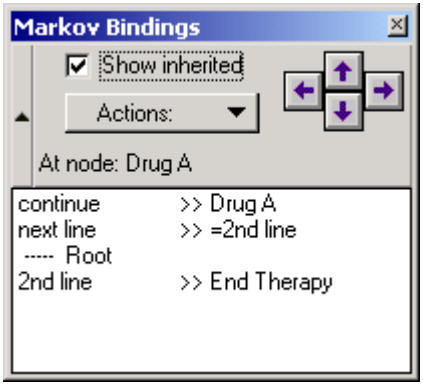
The use of Markov bindings can be illustrated using the Markov Bindings tree, shown below. A copy of the tree, which is unfinished, can be found in the Tutorial Examples\Healthcare sub-directory.



Transitions must be assigned to the Response and No Response nodes in the clone master. The Response transition node in Drug A's subtree should jump back to Drug A, while the same node in Drug B's clone copy subtree should, instead, jump to Drug B. Similarly, Drug A's No Response node should point to Drug B, while Drug B's No Response node should point to End Therapy.

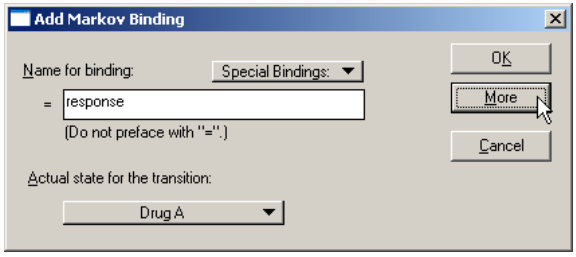
► **To define Markov bindings in the Markov subtree:**

▷ Open or activate the Markov Bindings tree. Select the Drug A node, choose **Options >** Show Markov Bindings Window….



In TreeAge Pro 2007, the Markov Bindings list is now displayed in a dockable window which works much like the Variable Definitions window introduced in Chapter 14. The Markov Bindings window includes navigation arrows and an Actions menu button.

▷ In the Markov Bindings window, Choose **Actions >** Add New….



▷ Type `response` in the Name text box, and specify a transition to Drug A by selecting it from the list of states. Click **More** to create a second binding called `no response` which points to Drug B.

▷ At the Drug B node, create two bindings called `response` and `no response`; specify transitions to Drug B and End Therapy, respectively.

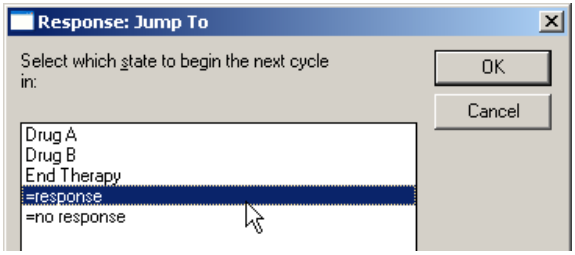
If the display of Markov information is turned on in the tree (in the Preferences dialog, under the Variables Display category), Markov bindings will be displayed below other Markov information, in the form *binding name* >> *jump-to state name*.

Bindings that you create have no effect until a binding name is referenced at a Markov transition node in a clone master.

► **To use a Markov binding in a clone master transition:**

▷ Select Drug A's Response node, and choose **Options >** Markov Jump State… (or double-click on the triangle node symbol).

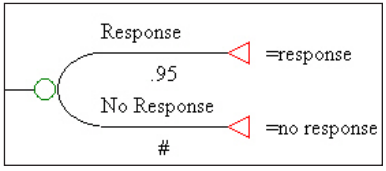
The Jump To dialog appears. In addition to the regular state names, the list now includes the binding names TreeAge Pro finds in the path back to the Markov node. The binding names displayed in the list are prefixed with the equal sign (=) to distinguish them from actual states. (Thus, when naming states, you should avoid using a leading =, although this is not strictly forbidden.)



▷ Select =response as the transition for Response, the currently selected node, and Press OK.

▷ Select the No Response node of Drug A, choose Options > Markov Jump State… and select the binding called =no response. Press OK to close the dialog.

When a binding name is used at a transition node, the search for the binding proceeds in right-to-left fashion, as with variables.



Markov Binding window notes:

- The **Actions** menu in the Markov Bindings window can be used to copy and paste lists of bindings (*name* >> *state* pairs) from one node to another, as well as into other files or a text editor. This allows you to set up a list of binding names and bind-to states in columns in Excel, and then copy into TreeAge Pro.
- The Markov Bindings window can be used to create bindings to the left of Markov nodes if necessary (e.g., if cloning entire Markov models). To add bindings outside of a Markov process, use the **Actions** menu to cut/copy/paste bindings as text.

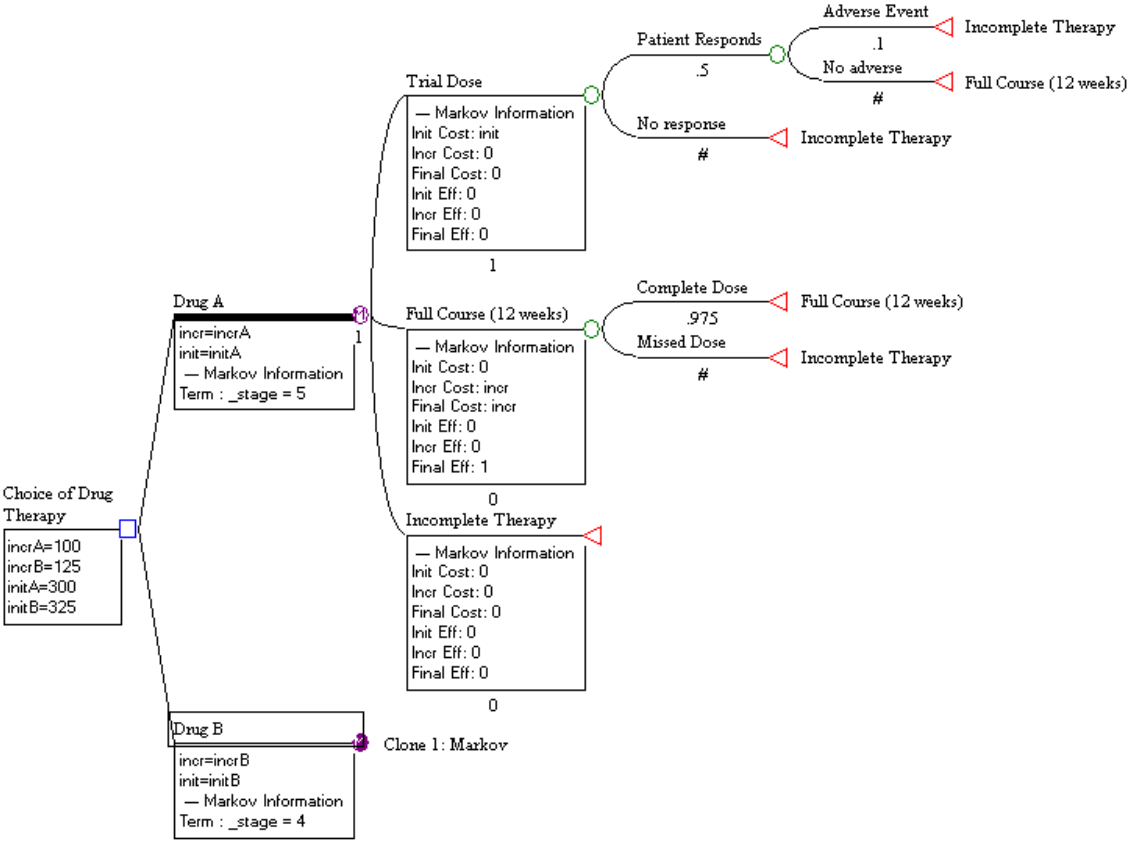
Tip:

To avoid Markov structure errors due to subtle differences in state names, you should use caution when trying to clone **part** of one Markov process in order to attach copies in a **different** Markov process. (If you are determined to try, however, TreeAge Pro will allow it!) You might instead try cloning the entire Markov process, creating a clone master at either the Markov node or a node to its left. There should be no problems, however, with attaching clone copies from one state onto another state within the same Markov process.

Cloning an entire Markov process

Consider the example Complex Markov tree, found in the Tutorial Examples\Healthcare subdirectory. A similar model, but using clones, is shown below. In the clone version of the tree, Drug B's transition subtree is a clone copy of Drug A's subtree. On the surface, these subtrees appear to be identical; in fact, the strategies have different termination conditions (assigned at the Markov node, *outside* the clone master).

State rewards utilize the variables *init* and *incr*. These variables are defined differently at the two Markov nodes, resulting in different cost calculations. Similarly, different values could be used for each subtree's probabilities, simply by converting the numeric probabilities to variables in the clone master, and then uniquely defining the variables at both the Drug A and Drug B Markov nodes.



Counting "time in state" using tunnels

A tunnel state can be used when you need to keep track of the number of cycles an individual has remained in a particular state. In a cancer state, for example, transition probabilities to other states often depend on how long the individual has been in the cancer state.

Note:
Instead of the tunnel state feature, some TreeAge Pro Markov models instead make use of user-defined *tracker variables and microsimulation* to count time in state. Each approach has possible advantages. Tunnel states work during both cohort analysis (i.e., rollback) and microsimulation, unlike trackers (which require microsimulation). Trackers, on the other hand, are extremely flexible, and can also be used to keep track of continuous variables, transient events, etc. Models built using tracker variables can be made structurally simpler, with fewer states/branches. See Chapter 36 for details on tracker variables.

Temporary states and the _tunnel counter

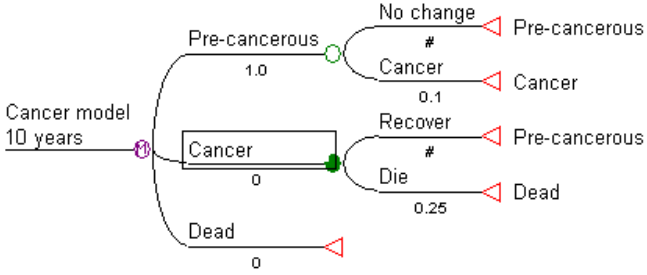
In the Markov modeling literature, a *temporary state* is a state which an individual must exit after one cycle, and a *tunnel* is a series of temporary states. Normally, an individual entering the tunnel state — either from another state, or at the start of the Markov process — enters temporary state #1. If an individual remains in the tunnel state for another cycle, they move in order through temporary states #2, #3, and so on.

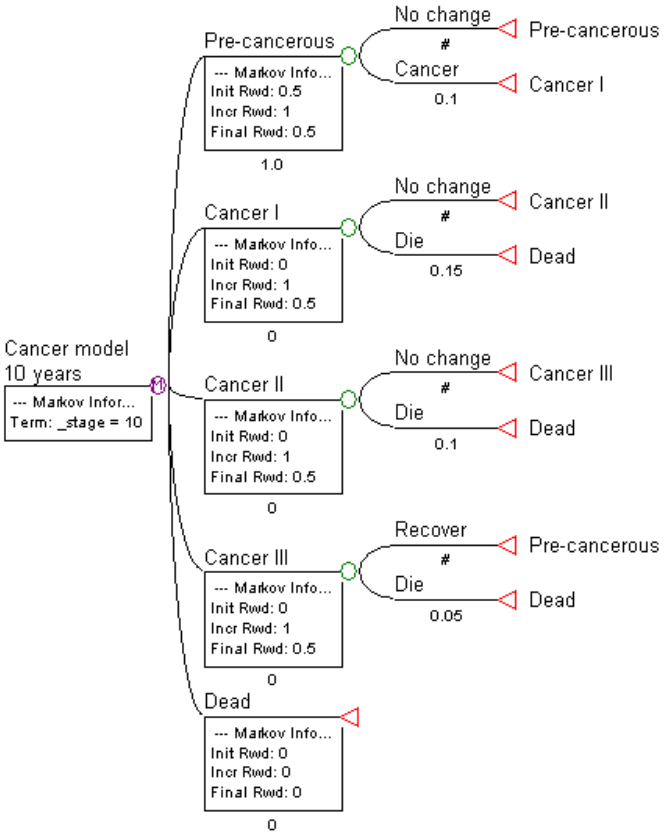
One way to model a tunnel is to use a separate state for each temporary state, and manually set up ordered transitions between the states. However, TreeAge Pro allows you to represent a tunnel more efficiently, using a single branch from the Markov node; this makes it easy to create tunnels of any length (even with thousands of temporary states).

When you create a tunnel state in TreeAge Pro, all temporary states will use the same transition subtree. In order to specify different transitions for particular temporary states, you can refer to TreeAge Pro's temporary state counter, a Markov keyword called **_tunnel** (similar to _stage). TreeAge Pro starts the _tunnel counter at 1 for someone entering a _tunnel state, and increments the counter by 1 each cycle they remain in the tunnel state. Using _tunnel to count "time in state," your transition probabilities can look up appropriate values from tables. Logic nodes and statements can also use the _tunnel counter.

Using a tunnel state – an example

Consider the three-state cancer model shown below. Note that since there is no transition from Cancer back to itself, the patient spends one cycle in the Cancer state before exiting to another state, and therefore Cancer is a *temporary state* (although not a tunnel yet).



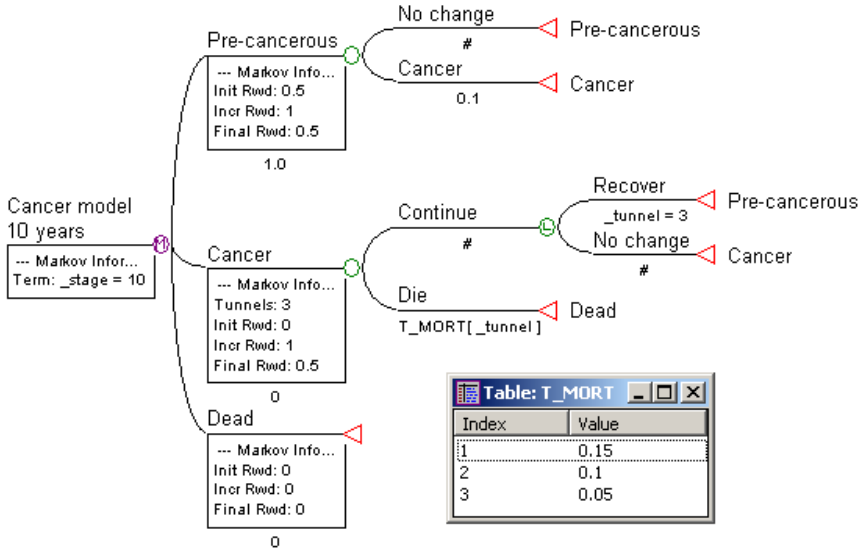


This model may not accurately represent the basic process of the disease. Cancer should probably unfold in a series of temporary states, with different probabilities of changing state (i.e., remission or death) in each successive cycle/year. And these probabilities should depend on how many cycles someone has spent in the Cancer state (which is not given by the _stage counter, since everyone starts in Pre-cancerous, and a transition to Cancer may occur at any cycle).

An "exploded" version of the cancer Markov model is shown at left. An explicit chain of temporary states is used to describe each year of the cancer. While this more detailed model is still relatively small, as the required number of temporary states increases, explicitly representing all of them becomes more problematic. In TreeAge Pro, the chain of temporary states can be represented more efficiently using a single tunnel state.

The same Markov process (i.e., calculating the same results) can been built in TreeAge Pro with Cancer set as a tunnel state, as illustrated below. (See the Cancer Tunnel.pkg file, including the required table, found in the Tutorial Examples\ Healthcare subdirectory.)

If the exact number of temporary states required is difficult to calculate (or is dynamic), err on the side of excess. Setting the number of temporary states too high has no adverse effects other than to create empty columns in reports. Later, after analyzing the model, you can reduce the number of temporary states to a more reasonable value.

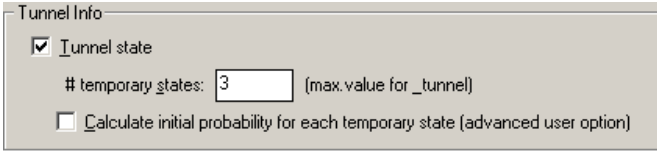


► **To change a state to a tunnel state:**

▷ Select the appropriate Markov state and choose **Values >** Markov State Information….

▷ Click the checkbox labeled Tunnel state.

▷ Enter the maximum required number of temporary states and click OK.

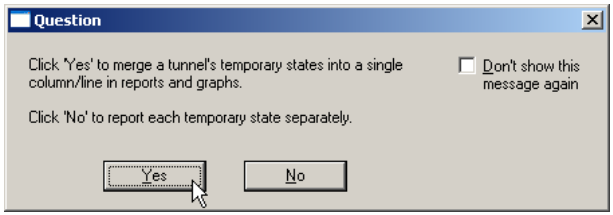


The number specified determines how high the _tunnel counter will increment (corresponding to the number of copies of the state which TreeAge Pro keeps track of internally, during calculations). Individuals that reach the last temporary state and transition into the state again will simply remain in the last temporary state.

Using the _tunnel counter

Note the use of the _tunnel counter/keyword in probabilities in the Cancer transition subtree. The Die node refers to a row in the T_MORT probability table using _tunnel, while the probability of the Recover node is equal to 1 if _tunnel = 3, and 0 otherwise.

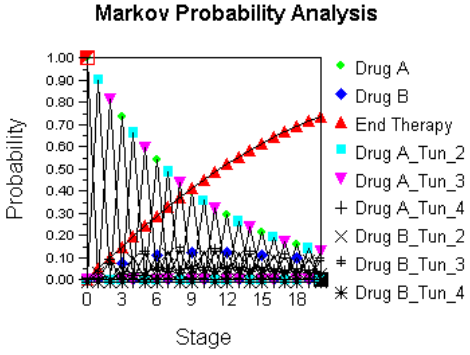
Merging results for temporary states

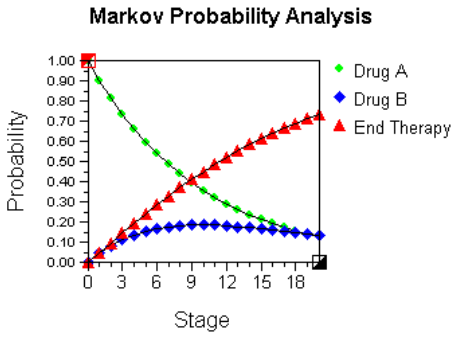

Question
Click 'Yes' to merge a tunnel's temporary states into a single column/line in reports and graphs.
Don't show this message again
Click 'No' to report each temporary state separately.
Yes
No

When you perform a Markov cohort analysis at a Markov node which includes a tunnel state, you are given the option to merge each tunnel's temporary states into a single text report column or graph line.

This is particularly useful if there are many temporary states in a tunnel. Prior to running the analysis, TreeAge Pro will present the prompt shown on the previous page, asking whether or not to merge temporary states. If you choose *not* to merge temporary states, the text report will append extra columns for the second and subsequent temporary states.

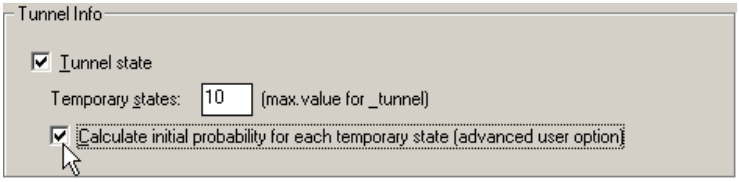
One positive result of merging a tunnel's temporary states is a simpler, more coherent line graph, as shown below.





Populating temporary states at cycle 0

Normally, the initial probability expression assigned to a tunnel state is used to populate only the first temporary state. It is possible, however, to distribute members of the cohort among the different temporary states at cycle 0, if necessary. The Tunnel Info section of the Markov State Information dialog includes an advanced setting that will cause TreeAge Pro to evaluate the initial probability expression in a tunnel state for every temporary state.



This could be used, for example, to create a model that uses the _tunnel counter to track the age of members of cohort with a realistic age distribution. Every state could be made a tunnel state, with their initial probabilities referencing tables of probabilities using the _tunnel counter (creating the age distribution).

Special binding names and tunnels

The original release of TreeAge Pro 2004 added a special "Tunnel Crossover" binding name; creating a binding with this name and using it at a transition node allows the **_tunnel** counter to increment uninterrupted when transitioning from one state to a different state, if both are tunnels. Normally, the _tunnel counter would reset to 1 when moving to a different state. (For more on creating and using Markov transition bindings, see the section on bindings later in this chapter.)

TreeAge Pro 2005 added two more special binding names: "Tunnel Variable" and "Tunnel Table". A binding with the name "Tunnel Variable" can be used to dynamically point any transition node to a specific temporary state in the destination tunnel state.

IMPORTANT!

Versions prior to Release 2004.0.7 do *not* recognize the Tunnel Variable and Tunnel Table bindings as being special; models using them will still calculate, but will not use the special transition behavior described here. (Prior releases of TreeAge Pro do, however, recognize the special Tunnel Crossover binding.)

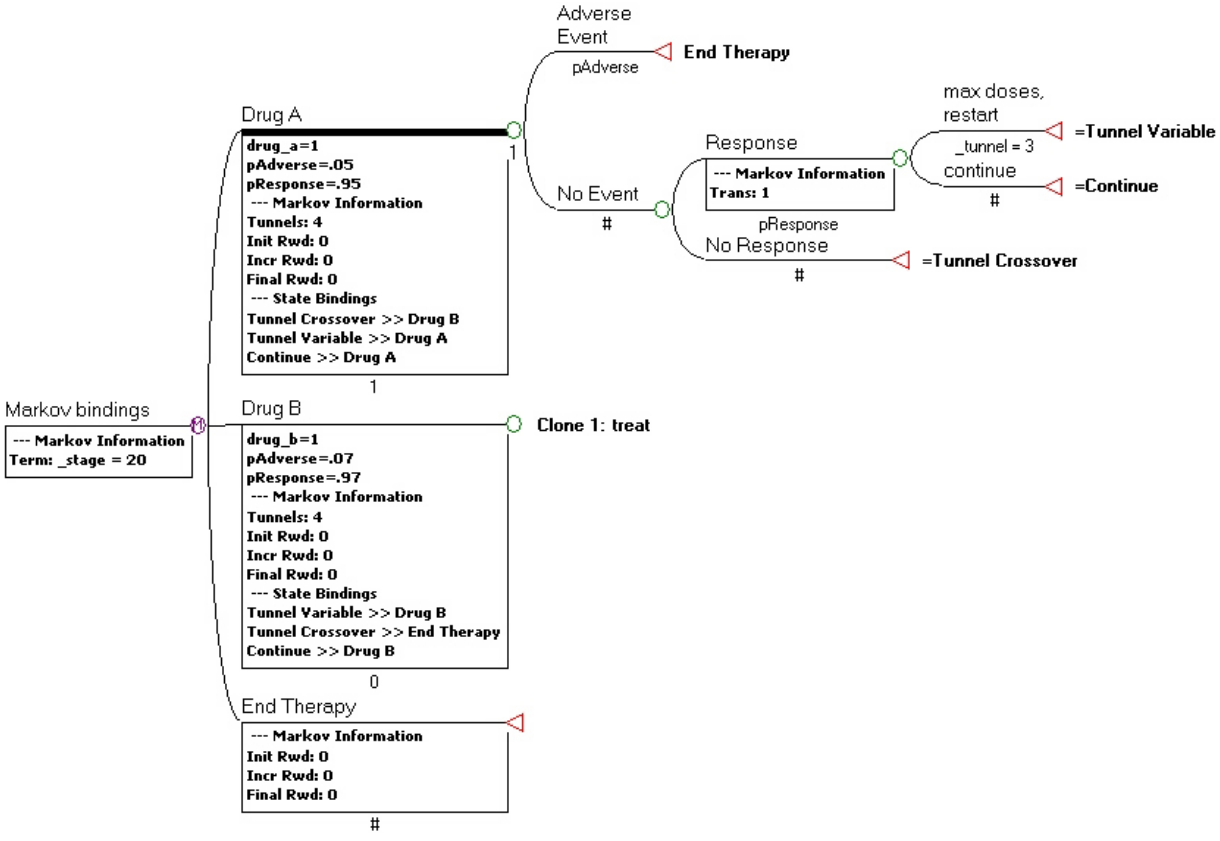
As noted above, transitions into a tunnel state normally enter in temporary state 1. Cohort members remaining in the state would normally progress through the temporary states in order, from 1 to 2 to 3, etc. The variable- and table-type transition bindings, however, can make jumps into, within, or between tunnels more flexible.

The example shown here illustrates the use of "Tunnel Variable" and "Tunnel Crossover" type bindings. A copy of this tree file, called Markov Tunnel Bindings, can be found in the \TreeAge Pro\ Tutorial Examples\Healthcare\ sub-directory.

The Tunnel Variable binding definition made at the Drug A node points to the Drug A state. The use of this binding at the "max doses, restart" node means that cohort members reaching this transition will remain in the Drug A state.

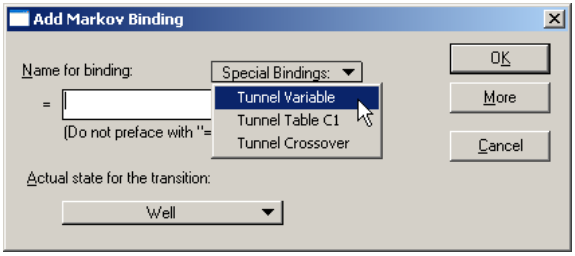
When encountered during transitions, however, the Tunnel Variable binding triggers a search for a variable — called "tunnel_Drug_A", in this case. The required variable name is based on the current state's name, adding the prefix "tunnel_", and converting to a variable name (substituting "_" characters for punctuation or other non-alphanumeric characters, and shortening to 32 characters).

In the example, the variable tunnel_Drug_A is defined equal to 1 at the Drug A node (the definition can made at any node, just as with other variables). Rather than automatically jumping to the next higher temporary state, the variable's calculated value of 1 will force cohort members reaching this transition to jump to the first temporary state (i.e., restart).



Note the use of a "Tunnel Crossover" binding in the example, pointing to the Drug B state from the No Response node in the Drug A state. Cohort members in the Drug A tunnel that reach this transition jump to the next higher temporary state in the Drug B tunnel. Also note the Drug B state's use of a clone of Drug A's subtree. All bindings are redefined to the left of the clone copy. The Tunnel Variable binding will search for a variable called "tunnel_Drug_B" in this case, matching the current state's name.

With the new tunnel binding options, a "Special Bindings" pop-up menu has been added to the Add Bindings dialog, listing the "Tunnel Crossover", "Tunnel Table", and "Tunnel Variable" options.



The special "Tunnel Table" binding name can be used similarly to "Tunnel Variable", with transitions to particular temporary states determined by numbers pulled from a column in a TreeAge Pro table named after the tunnel state. With a Tunnel Table binding, the current value of the _tunnel counter is used to select a row from the table in order to retrieve the number of the temporary state to jump to. If a multi-column table is used, the binding name can be changed to "Tunnel Table C*x*" where "*x*" is the number of the value column to use for the jump-to temporary state index. (TreeAge tables support up to 512 value columns.)

Tips:

The _tunnel keyword can be used in the definition of the tunnel binding variable. Define drug_a equal to "_tunnel + 1", for example, would result in the default behavior (jump to next higher temporary state).

Modifying a tunnel state's behavior using special bindings offers an efficient way to set up a Markov model with effectively hundreds or thousands of states. The visible tree structure could be relatively small, using a "template" transition subtree. The complexity would be contained within the variables and tables used in determining transitions between "temporary states," as well as probabilities and rewards.

Markov decision processes

The TreeAge Pro Healthcare module is able to dynamically evaluate decision nodes within Markov processes during expected value calculations, including 2nd-order Monte Carlo simulation (not during microsimulation).

Why use decisions in a Markov cohort analysis?

Within a Markov process, TreeAge Pro is able to make decisions based on "future" expected values — i.e., by comparing the net expected state and transition rewards for the time remaining in the Markov process starting at the state (and cycle) where the decision occurs. The Markov decision process is evaluated in two steps.

First, future expected values are determined by working backwards in the model (starting with final rewards). For this reason, a basic requirement of Markov decision processes is a termination condition that is a predetermined number of cycles to run, *N*. TreeAge Pro calculates the final reward for each state at cycle *N*, allowing decisions to be made in the previous (next-to-last) cycle, *N*-1. For a particular decision, the optimal policy's future value is combined with the state reward for cycle *N*-1, making it possible to make decisions and calculate future expected values in cycle *N*-2, and so on.

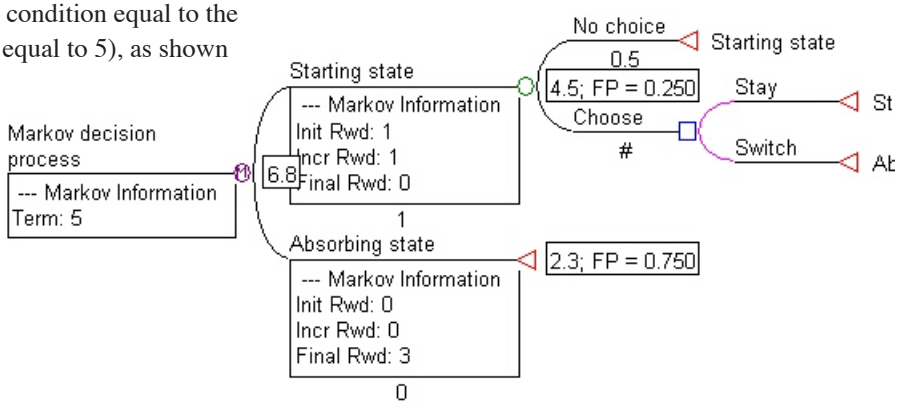
Transition rewards ("tolls") that occur at or to the left of a decision in a particular cycle will not influence the decision in that cycle; tolls in branches of the decision node (or to the right) will influence decisions (in combination with the future rewards of the jump-to states).

Once optimal policies have been determined for every decision at every time point by working backwards to cycle 0, the next step is to perform a standard cohort analysis (forwards). During the cohort analysis, the portion of the cohort that reaches a particular decision node at a particular cycle will follow the optimal path determined in the "roll back" step, and continue with transitions.

Implementation — special termination condition

If you have a decision node to the right of a Markov node, TreeAge Pro requires a special, ***numeric termination condition***. This is required so that the cohort/EV analysis can know the number of cycles prior to the analysis. You cannot use a logical test, such as "_stage=20", as you would in a regular Markov process.

To run a Markov decision process for 5 cycles, simply set the termination condition equal to the number "5" (or a variable equal to 5), as shown below.

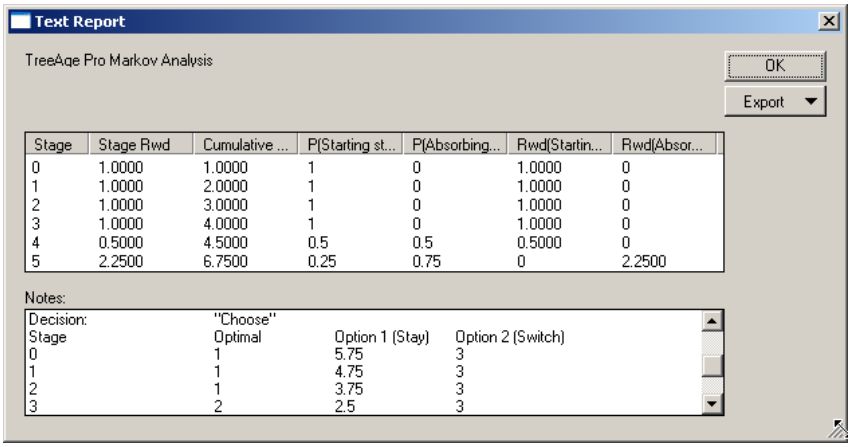


Markov decisions reporting

If a roll back analysis is performed, TreeAge Pro identifies policies in a Markov decision process that are optimal at any cycle with magenta branches; policies that are never optimal are hashmarked. The Markov model shown below illustrates. A copy of this tree file, called Markov Decision Process, can be found in the \TreeAge Pro\Tutorial Examples\ Healthcare\ sub-directory.

In this simplistic case, at different cycles "Stay" or "Switch" are found to be optimal. The logic of the decision making process is that there is only a 50/50 chance at each cycle of being able to choose a state for the next cycle. The analysis determines at what cycle it is optimal, if given the choice, to switch to the absorbing state. (Switching too soon means sacrificing the incremental reward of 1 each cycle; waiting too long means potentially missing out completely on the chance to switch to the absorbing state and claim the final reward of 3.)

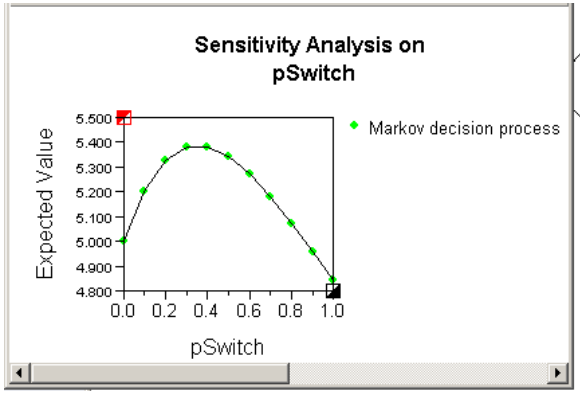
If a cohort analysis is performed at the Markov node, the Notes section of the text report will include a state-by-state, cycle-by-cycle policy analysis, for each decision.

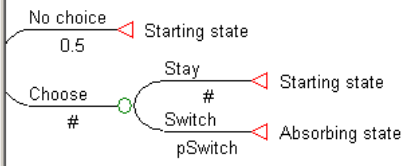


Stage	Stage Rwd	Cumulative ...	P(Starting st...	P(Absorbing...	Rwd(Startin...	Rwd(Absor...
0	1.0000	1.0000	1	0	1.0000	0
1	1.0000	2.0000	1	0	1.0000	0
2	1.0000	3.0000	1	0	1.0000	0
3	1.0000	4.0000	1	0	1.0000	0
4	0.5000	4.5000	0.5	0.5	0.5000	0
5	2.2500	6.7500	0.25	0.75	0	2.2500

Decision:	"Choose"		
Stage	Optimal	Option 1 (Stay)	Option 2 (Switch)
0	1	5.75	3
1	1	4.75	3
2	1	3.75	3
3	2	2.5	3

In the example, note that without the decision node — if a chance node was used instead, or a single strategy was chosen before the process — then the expected value of the process would always be less than the 6.8 calculated by the decision process. A sensitivity analysis in the non-decision model shows a maximum value of 5.4.





Other types of expected value analysis (including Monte Carlo simulations that *do not* use trials) will do the future value analysis for each Markov decision process and then calculate the expected value for the Markov node following the per-cycle optimal paths; these analyses will not explicitly report the optimal paths.

Modeling discrete and/or changing size "populations" during cohort analysis

Normally, decision trees calculate an expected value for an average project, individual, or other unit. In budget-oriented or population-based modeling, however, the goal often is to determine not an average, but an overall cost or benefit. In some cases this is done simply by multiplying the regular expected value by some number (e.g., the number of projects, or the size of a population). The multiplication might occur outside of the tree, or every payoff or reward could be multiplied by a variable representing this number.

An equivalent option would be to do the multiplication via tree probabilities. This would require that, in every path, a single probability be multiplied by the population size.

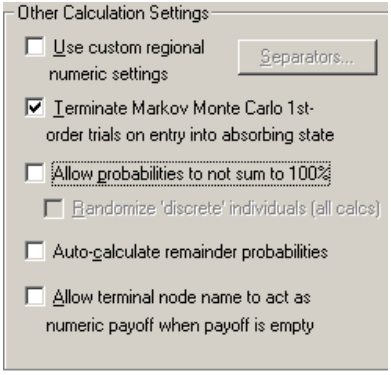
It has been noted numerous times in this manual that TreeAge Pro requires that the branch probabilities of each chance and Markov node in your tree always sum to 1.0. Probabilities that meet this requirement are referred to as "coherent." However, in situations where it is useful to do so, this restriction can be removed.

TreeAge Pro includes a tree preference to turn off the error checking that normally protects against mistakenly assigning non-coherent probabilities.

It is not recommended that the option to allow non-coherent probabilities be used without careful consideration of its appropriateness and the hazards of turning off probability error checking.

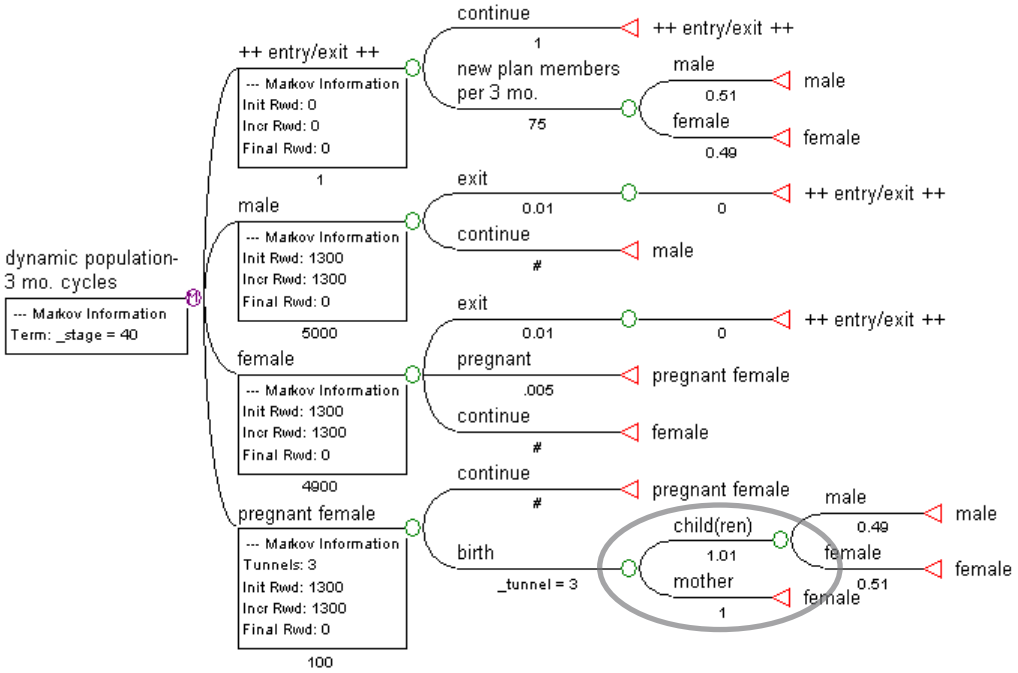
► **To disable errors when using non-coherent probabilities:**

▷ Choose **Edit >** Preferences…, and select the Other Calc Settings category.


Other Calculation Settings
Use custom regional numeric settings
Separators...
Terminate Markov Monte Carlo 1st-order trials on entry into absorbing state
Allow probabilities to not sum to 100%
Randomize 'discrete' individuals (all calcs)
Auto-calculate remainder probabilities
Allow terminal node name to act as numeric payoff when payoff is empty

▷ Check the option labeled Allow probabilities to not sum to 100%.

A simple example, the Dynamic Population model from TreeAge Pro's Tutorial Examples\Healthcare subdirectory, is shown on the following page.



This model illustrates the use of non-coherent probabilities to model a growing population (e.g., for budget impact analyses). It highlights two key aspects of using non-coherent probabilities in Markov models:

- discrete initial sizing of the cohort (e.g., N=35 million) using non-coherent ***initial*** probabilities
- population growth (e.g., through births or migration) using non-coherent ***transition*** probabilities

In some models, just setting a size on the initial cohort (i.e., number of individuals starting in each state instead of initial probabilities) might be useful, in combination with the StateProb( ) function, described in this chapter. A state "probability" would actually be a number of individuals. In an infectious disease model, the probability of a new infection could depend on the number in infected states:

```
pInf = .2*StateProb(6;10)/StateProb(1;10)
```

In the Dynamic Population example, only population growth is modeled (not disease). The context is a health plan with changing membership. The starting membership is specified with numbers of individuals starting in the states — a total of 10,000 individuals in the example (ignore the "++ entry/exit ++" state). During 10 years of Markov calculations (40 three-month cycles), individuals can be added to the population in two ways:

- entry from other populations, outside the health plan (i.e., new enrollment)
- internal population growth (i.e., births and other changes in family size)

The entry/exit state is manipulated to always have a state probability of 1. This makes it easy to incorporate new enrollees (with transition "probability" of 75 new members per cycle in the example). Births to plan members, highlighted above, uses non-coherent transition probabilities (1.01+1). Population loss is modeled: note the transitions to entry/exit with 0 probability, used to make individuals disappear from the model (rather than using an absorbing state).

Notes on dynamic-sized cohorts:

- As a precaution against unintentional use of the non-coherent probabilities setting, TreeAge Pro will not use this as a default preference for new trees, and will prompt you to reconfirm this preference each time a tree is opened.
- Non-coherent probabilities are not currently compatible with Monte Carlo microsimulation. Consider using the Node() function instead, to link to another Markov process to run microsimulation trials.
- A sub-option available when allowing non-coherent probabilities is to maintain integer "probabilities" at chance nodes — in effect, keeping individuals whole by randomizing each based on chance node probabilities. This might be useful, for example, if small probability events are critical (i.e., in a model where continued transmission of a contagious disease requires at least one carrier). This option will only work in Markov models with coherent initial and transition probabilities — non-coherent probabilities can instead be used in chance nodes to the left of the Markov node.
- Instead of having TreeAge randomize whole individuals as described above, you can selectively and carefully use the rounding functions (Floor, Ceiling, or Round) to the same effect.
- If non-coherent probabilities are used not to model a finite-sized population, but instead to enable non-exclusive chance node branches — i.e., to create the possibility of going down more than one path — care must be taken not to double-count payoffs.

Calculations using the StateProb function

TreeAge Pro's StateProb() function can be used to returns the state probability at the beginning of the current cycle for either the state currently being calculated, or for other states:

- If the StateProb() function is used without arguments it calculates the state probability for the current state (e.g., if you are calculating a transition probability in a state that somehow depends on the current probability of that state).
- To return the path probability of a different state, include a single argument equal to the state's branch index starting at 1 for the top state/branch – e.g., StateProb(1).
- To return the sum of the state probabilities of a range of states, specify the range of branch indexes using two arguments, for example StateProb(2;5) to add the state probabilities of states 2 through 5.
- The StateProb() function will work with non-coherent probabilities in a dynamic cohort model, as described on the previous page.
- The StateProb() function will also work during microsimulation if parallel trials are used; see Chapter 36.

TreeAge Pro also includes a PathProb() function, which calculates the path probability of the node being calculated. In a Markov model, however, PathProb() returns the path probability of the Markov node. TreeAge Pro 2007 adds another function, BranchProb(), which returns the calculated probability for a chance node branch.

Nesting or linking Markov models

TreeAge Pro includes powerful functions – Node(), Global(), GlobalN(), and User(), for example — that can be used to do things like nest one Markov model within another, link many terminal nodes to one Markov model, or combine a Markov microsimulation with expected value analyses like 1-way sensitivity analysis. Function syntax is described in more detail in Chapters 14 and 21 of the manual.

A trivial example illustrating the use of these two functions is shown below. The top arm is set up for expected value calculation, and its decision node includes two strategies both linked via the Node() function to the same Markov model at the bottom. Each strategy passes values to the Markov process using the Global() function. The Node() function can calculate the Markov model using microsimulation trials (even during non-simulation analyses of the top decision node).

The Node() function's arguments are used to select a node in the tree and to determine what kind of calculation to use at that node. The syntax is:

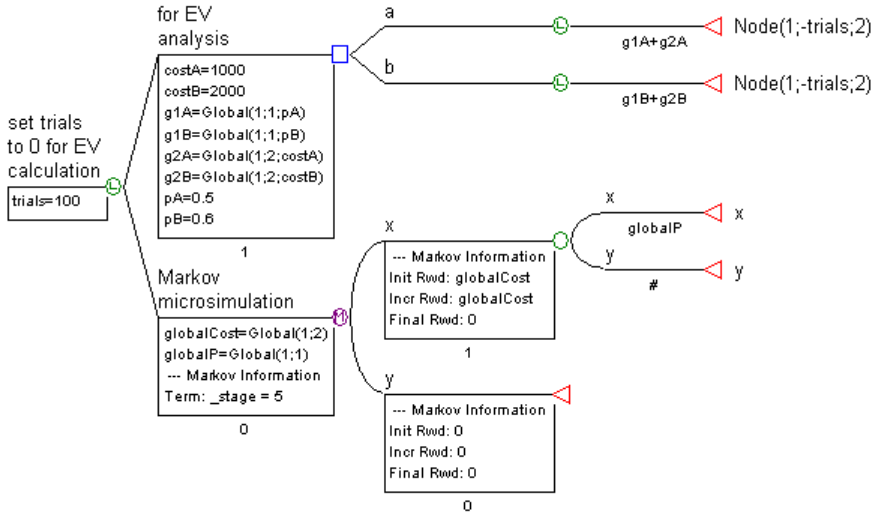
Node(*attribute*; *method*; *branch*; …)

In CE models the *attribute* argument determines whether to return cost or effectiveness (-1 calculates and returns cost; -2 returns effect from previous calculation; -11 returns cost from previous calculation; -12 calculates and returns effect). In non-CE models, as in the example, use any number other than 0 (which returns 0).

For the *method* argument use 0 to calculate the expected value; or specify a negative number to average that number of microsimulation trials (e.g., -100 to run 100 trials), as in the example.

The third and subsequent arguments are branch numbers used to select a node, starting with a branch of the root node.

The Global() function is illustrated in the example, but is not necessary for linking Markov processes. It can be used to store transient values, such as a snapshot of the value of trackers used in a subsidiary Markov process.



The following syntax is used to calculate and store a value in a cell in the Global matrix:

Global(*row*; *column*; *expression*)

The function returns the value of the calculated expression, as well as stores it in the Global matrix. The first cell in the global matrix is at row=1, column=1. Up to ten thousand cells are currently supported.

The following syntax can be used to reference a value saved to the global matrix:

Global(*row*; *column*)

The contents of the global matrix can be dynamically saved to a text file (or emptied) using the third syntax of the Global() function:

Global(*value*)

If value evaluates to a non-zero number, the contents of the global matrix are silently saved to a text file in the tree's directory. A zero value will empty the matrix. See Chapter 21 for more information.

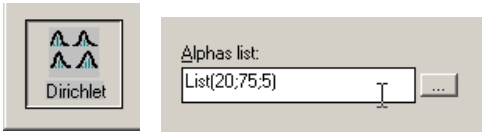
Notes on the Node() function:

- Trackers are not reset to 0 at the beginning of trials run by the Node( ) function. This is intentional, to allow communication between the calling and called nodes.
- Trials run by the Node( ) function do not resample distributions automatically (even if their properties are set to sample per trial). To force a sample, use the Dist(N; 1) syntax or DistForce(N). To control the sampling rate, put the force sample expression in a tracker evaluation (outside a Markov process, for example).
- The second parameter, specifying a number of trials, can be given a number after a decimal place to indicate that a statistical measure other than the mean should be used from the set of trials.
- Function syntax is described in more detail in Chapters 14 and 21 of the manual.

Sampling probabilities from a multivariate Dirichlet distribution

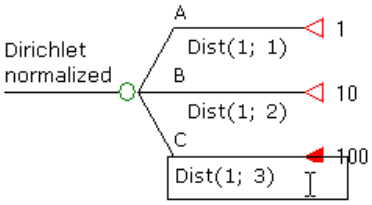
If a chance node has more than two branches with non-negligible probabilities, performing a sensitivity analysis or Monte Carlo simulation that changes the values of these probabilities can be problematic. One option is to normalize the chance node's probability expressions. For example, if a node has three outcomes, A, B, and C, rather than assigning variables to two probabilities and using the # remainder for the third, you could do the following: assign three expressions that always sum to 1.0, like `pA/(pA+pB+pC)` and `pB/(pA+pB+pC)` and `pC/(pA+pB+pC)`. No matter what values (>= 0) are assigned to pA, pB, and pC, the three normalized probabilities will always sum to 1.0. (The # remainder could still be used in place of one of these.)

TreeAge Pro offers a similar solution using a special, multivariate form of the beta probability distribution, called a Dirichlet distribution. This distribution can be used represent the uncertainty in all of the probabilities of a chance event. During Monte Carlo simulation, the distribution can sample probabilities for each branch, using normalization to ensure that the probabilities always sum to 1.0.

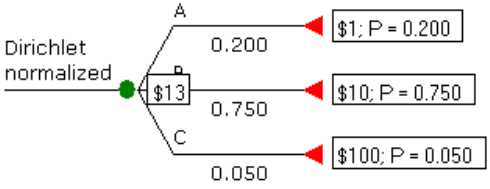


If the distribution is parameterized with a list of three positive *alpha* values, as shown here, TreeAge Pro will samples three independent Gamma[*alpha*, beta=1.0] distributions and normalize these to create a list of three probabilities.

To utilize the generated sample probabilities, the Dist() function described in Chapter 16 is used with a second argument to indicate which branch (i.e., alpha) to use, as shown below.



Rolling back the tree shows the mean values of the probabilities, which are simply the normalized alpha parameters.



Performing a simulation in the example tree shows the effect of sampling independent Gamma distribution values, based on the list of alpha parameters, and then normalizing. For each iteration of the simulation, a different set of Gamma random variates is drawn. Each iteration results in a different sum, as well as different ratios of the Gamma random variates to the sum (i.e., the probabilities), but normalization ensures that the resulting probabilities sum to 1.0.

> ***Note:***
> Simulation text report and graphs only report the first sampled Dirichlet probability. As an alternative for reporting the multivariate sample values, either tracker variables or the special Global() matrix function can be used.

Markov cohort analysis using the TreeAgeProLib or TreeAge Pro Interactive

Users of the TreeAge Pro Excel/COM module, or TreeAge Pro Interactive, can create macros or other automation scripts/programs that run Markov analyses programmatically. For example, the macros would use a TreeAgeProLib.ApplicationObj variable to create a TreeAgeProLib.TreeObj variable, and then a TreeAgeProLib.MarkovOutput variable.

The advantages of the programmatic approach to running the Markov cohort analysis include:

- Automate repetitive analyses
- Parse sections of the full text report for specific values
- Pause between cycles in order to make complex adjustments to the model (using MarkovEvents)

See Chapter 30 and Appendix F for more information on the TreeAgeProLib library, COM automation, and related features.

Chapter 36

Markov Monte Carlo Microsimulation

This chapter covers the use of Markov Monte Carlo microsimulation with the TreeAge Pro Healthcare module, including two-dimensional simulation, tracker variables, per-trial sampling distributions, and parallel trials.

Chapter 33 covers Cost-Effectiveness simulation outputs.

See Chapter 16 for general instructions on using Monte Carlo simulation, and Chapter 17 on defining distributions for microsimulation models.

In this chapter:

- Microsimulation and tracker variables 502
- Debugging Markov microsimulations 507
- Two-dimensional probabilistic sensitivity analysis using microsimulation 508
- Resampling distributions during microsimulation trials 511
- Sequences and linear sensitivity analysis 513
- Other aspects of microsimulation 515

Microsimulation and tracker variables

As described in Chapters 34 and 35, in a cohort expected value analysis, a Markov model retains no memory of previous events from one cycle to the next. Transitions and rewards are assigned to portions of the cohort based only on their state membership in the *current* cycle. The portion of the cohort starting a given stage in a given state is treated as a homogenous group, with no regard given to the different paths that are possible prior to entering that state at that time.

In order to enable greater flexibility in Markov models, analysis can be done via Monte Carlo microsimulation trials, following one individual at a time through the model. Averaging the results of a large number of such individual trials can simulate the calculation of an expected value.

During microsimulation TreeAge Pro enables special variables called trackers, to track detailed patient history, for example. Microsimulation and trackers can also be used to build a discrete event simulation model. Rather than representing equal time intervals, each cycle simply represents the variable time to the next event of interest. A tracker variable can be used to actually track the passage of time (instead of using the _stage counter).

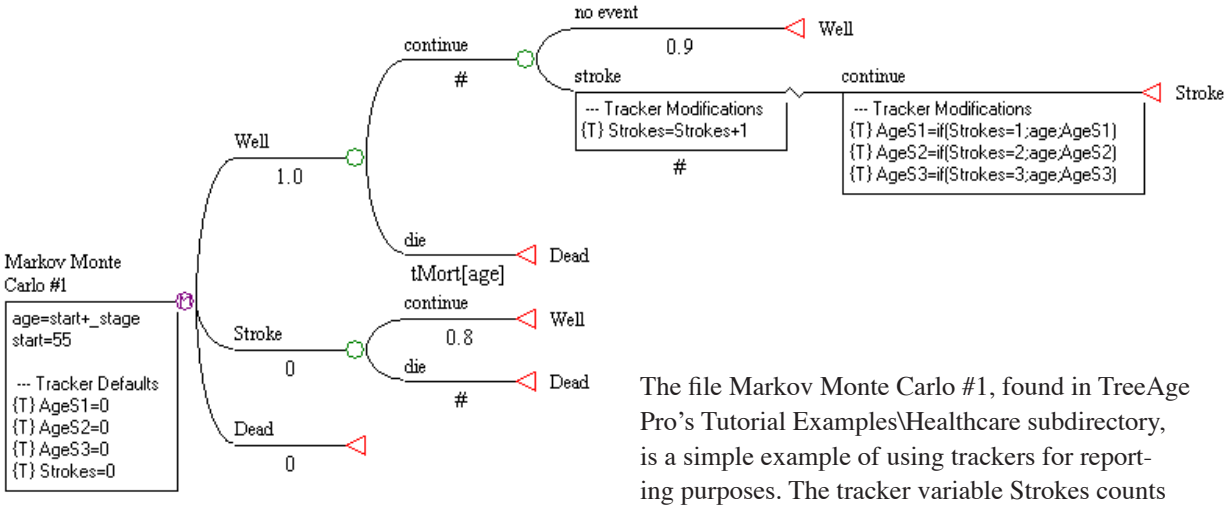
This section introduces the use of tracker variables in your Markov model, and the selection of appropriate settings for Monte Carlo microsimulation.

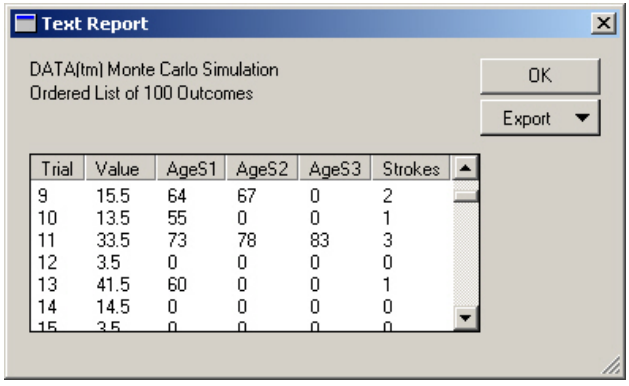
Using tracker variables for reporting

Tracker variables can be used for reporting additional output quantities (i.e., other than cost and effectiveness), such as the number of stages spent in a particular state.

Without the use of tracker variables, there is no practical way to build a model that can remember that even a few of the events a patient has suffered over time, such as how long ago a stroke occurred. With trackers, however, detailed memory of any prior event can be easily retained and recalled.

The current value of a tracker variable for a particular trial can be modified by redefining the tracker variable at the event node. If a microsimulation trial encounters a node with the tracker modification numEvents=numEvents+1, for example, TreeAge Pro will take the current global value of numEvents for that patient's trial, add 1 to it, and store the new value.





Trial	Value	AgeS1	AgeS2	AgeS3	Strokes
9	15.5	64	67	0	2
10	13.5	55	0	0	1
11	33.5	73	78	83	3
12	3.5	0	0	0	0
13	41.5	60	0	0	1
14	14.5	0	0	0	0
15	3.5	0	0	0	0

The file Markov Monte Carlo #1, found in TreeAge Pro's Tutorial Examples\Healthcare subdirectory, is a simple example of using trackers for reporting purposes. The tracker variable Strokes counts the number of times a trial patient enters the Stroke state. The other three trackers record the age of the trial subject at the time of successive strokes.

The model requires a probability table, tMort, created in the tutorial in Chapter 35. Create that table now, if you have not already done so.

The model does not include any sampling distributions, so the only type of Monte Carlo simulation that will be available under the Analysis menu will be microsimulation. Run 100 trials at the Markov node and look at the microsimulation text report.

At the start of each trial, all four trackers are reset to 0 (this initial value can be changed in the variable's Properties dialog, as described later). If at some cycle an individual enters the Stroke state, the tracker modification Strokes=Strokes+1 is executed, incrementing the global value of the Strokes tracker variable.

Note:

The tracker values reported for two-dimensional simulations, which sample distributions and then run microsimulation trials for each set of sample values, are averages for each group of trials run for a particular set of parameter samples.

Using trackers in Markov calculations

In addition to their uses in reporting, trackers can also be used in calculations of transition probabilities, rewards, termination conditions, and any other Markov expression.

For example, one set of tracker variables could be used in a model to indicate how often a patient has undergone a particular treatment, while another set of tracker variables keeps track of the current size, type, and location of a tumor. A logic node could compare the value of the tumor size tracker variable to some threshold and appropriately transition the individual to a new state. Or, the tumor location tracker variable could be used as a lookup value in retrieving an appropriate Markov reward from a table of surgical costs.

The example shown on the previous page, Markov Monte Carlo #1, uses trackers as outputs, only — the stroke tracker variables are not used in Markov calculations. Sensitivity analysis, Markov cohort analysis, and other expected value calculations can safely be used with this model – they just won't report tracker variable values.

The example files Markov Monte Carlo #2 and #3, on the other hand, illustrate how trackers might be used in Markov calculations. These models can be analyzed *only* with microsimulation trials; they will not calculate correctly unless tracker modifications occur.

Since expected value sensitivity analysis is not valid with Markov models using trackers as inputs, examining parameter uncertainty in such cases requires special forms of sensitivity analysis based on first-order trials. A later section in this chapter covers probabilistic sensitivity analysis using microsimulation.

Chapter 35 describes how the Node() function might be used to enable EV analyses for Markov models requiring microsimulation.

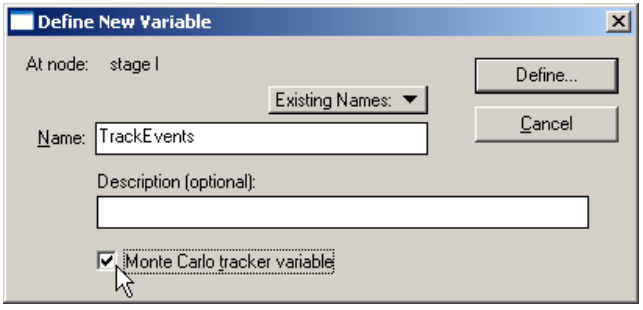
Creating and defining tracker variables

Tracker variables are similar to the regular variables described in detail in Chapter 12, except for the way in which each is evaluated during calculations. The set of tools outlined in Chapters 12–14 can be used to create, define, and manage tracker variables as well as regular variables.

The quickest way to create a tracker variable is by right-clicking on a node that requires a tracker modification.

► **To create and define a Monte Carlo tracker variable:**

▷ Right-click on an event node, usually within the Markov subtree, and choose **Define Variable >** New….

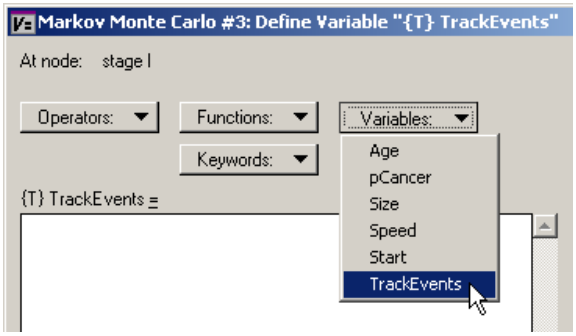


▷ In the dialog, enter a variable name and make sure to check the option labeled Monte Carlo tracker variable. Click OK.

Tracker variables are not usually given definitions in the same sense that normal variables are. Instead, tracker modifications usually look something like: TrackEvents = TrackEvents + 1

The expression on the right of the = sign is what is entered in the Define Variable window.

- ▷ In the Define Variable window, enter a tracker modification and click OK.



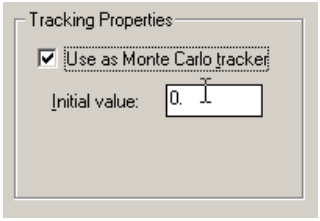
When TreeAge Pro displays a tracker variable's name in a dialog or window, it is prefixed with {T} to indicate that it is not a normal variable. Do *not* type this prefix when referring to the tracker in formulas.

To change (or delete) an existing tracker modification, simply right-click on the desired event node and choose the variable from the Define Variable list.

Tracker variable properties

Another way to create a tracker variable is using the Variables and Tables dialog. This allows you to change the tracker's initial value property to something other than the default 0, if necessary.

- ▷ Choose **Values >** Variables and Tables…. In the Define Values dialog, choose New > Variable….
- ▷ In the variable Properties dialog, name the variable and check the option labeled Use as Monte Carlo tracker box.



The initial value is the global value of that variable at the beginning of each trial. In many cases, the default value of 0 can be used.

- ▷ Optionally, you can assign a non-zero initial value for the tracker.
- ▷ Click OK to save the variable properties, and again to close the Variables and Tables dialog.

Tracker modifications that reference other trackers

If you have more than one tracker modification at a particular node, as in the example, the modifications will be applied in *reverse alphabetical order*. To avoid errors and confusion, if one tracker modification is *dependent* on another tracker modification, they should be separated. Define the dependent tracker modification at a successive node.

This can be accomplished by inserting a branch to the left or right of the existing event node, so that the input tracker modification occurs before the dependent tracker modification. The node between the two modifications can be changed to a label node, if its only function is to separate the trackers.

In the example shown on the previous page, the modifications of the AgeS*n* trackers are placed at a label node following the Stroke node.

Note that this is not a problem with a single, *self-referential*, incrementing tracker modification, such as the commonly-used form:

trackerX = trackerX + 1

Tracker modification timing

Tracker modifications at a node occur after rewards (state or transition) are accumulated at that node.

Trackers and expected value calculations

While expected value (EV) analysis is not recommended for trees in which tracker variables are referenced in probability or reward calculations, this type of analysis is not disabled. (Note that trees that use trackers solely as outputs, not as calculation inputs, are not subject to the limitations discussed here.)

Since tracker modifications are only meaningful within simulation trials, TreeAge Pro generally ignores them during EV calculations, *including Monte Carlo simulation EV/cohort calculations*. Outside of simulation trials, the values of trackers will be equal to their initialization value.

One exception to this rule is when an expected value analysis makes a call to the Node( ) function which in turn runs a set of microsimulation trials at another location in the tree. See the information on the Node( ) function in Chapter 35.

Another possible application of trackers in expected value calculations of regular trees is as a more efficient replacement for recursive variable definitions. Under Other Calc Settings, there is a preference that must be turned on for tracker modifications to work like recursive variable definitions during regular, non-microsimulation analyses. See Chapter 14 for information on recursive variable definitions.

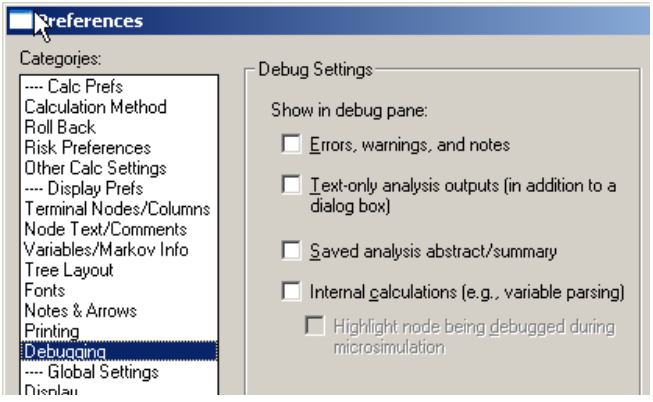
In light of the potential for error, it is advisable to limit analysis of trees using tracker variables to Monte Carlo microsimulation. For the purposes of EV analyses, like roll back or one-way sensitivity analysis, it may be desirable either to use the advanced Node() linking function, described in Chapter 35, or to develop a modified version of the model in which calculations do not depend on tracker variables.

Debugging Markov microsimulations

The Debug pane, described in Chapter 21 and Appendix C, can be used to display/report a variety of textual information, including error messages, text-only analysis output, stored analysis summaries, and detailed internal calculations (e.g., step-by-step evaluation of variables).

One way to turn on the output of debug information, go to Edit > Preferences, and change the settings under the Debugging category. This should be turned off again once you are done debugging, to speed up analyses.

A sub-option under Internal calculations, Highlight node being debugged…, is designed to allow the user to pause calculations at intervals during a microsimulation. For example, at each step in a random walk, each variable calculation, etc., the debugging output could be verified in the debugging pane, in order to check for calculation correctness in complex models.



Using the Debug() function and the Debug pane

If you turn on detailed calculation reporting in the tree's Debugging preferences, each probability, payoff, and other calculation is reported in detail. During Monte Carlo simulations, it reports each random walk and sampled value.

The Debug( ) function can be used to control the Debug pane output during analyses (or to add custom text to the pane). This is of primary interest in complex analyses, when the full debugging output of calculations may be hundreds of thousands of lines (or more) per analysis. Use the Debug( ) function to dynamically turn on and off the flow of calculation outputs to the Debug pane at strategic points in the analysis.

See Chapter 21 for details on the Debug( ) function syntax.

Two-dimensional probabilistic sensitivity analysis using microsimulation

Some Markov models require microsimulation, either for tracker variables or per-trial sampling distributions (discussed in Chapter 16 and later in this chapter). Performing sensitivity analysis on these microsimulation models will generally require probabilistic sensitivity analysis using Monte Carlo simulation (rather than n-way, expected value analyses).

The probabilistic sensitivity analysis method is to sample sets of parameter values and, for each set, recalculate the expected values for the model. Chapter 16 covers the general usage of Monte Carlo simulation to perform probabilistic sensitivity analysis.

The theoretical background for this kind of probabilistic sensitivity analysis is discussed in detail elsewhere, for example in papers published in the Journal of the Society for Medical Decision Making including:

> "Representing First- and Second-order Uncertainties by Monte Carlo Simulation for Groups of Patients," Elkan Halpern, Milton Weinstein, Maria Hunink, G Scott Gazelle, *Med Decis Making* 20:314-322 (2000).
>
> "Uncertainty in Decision Models Analyzing Cost-Effectiveness," Bruce Craig, Michael Black, Peter Sendi, *Med Decis Making*, Letter 20:134-136 (2000).

You are urged to explore these and other publications on this topic.

Performing sensitivity analysis with microsimulation

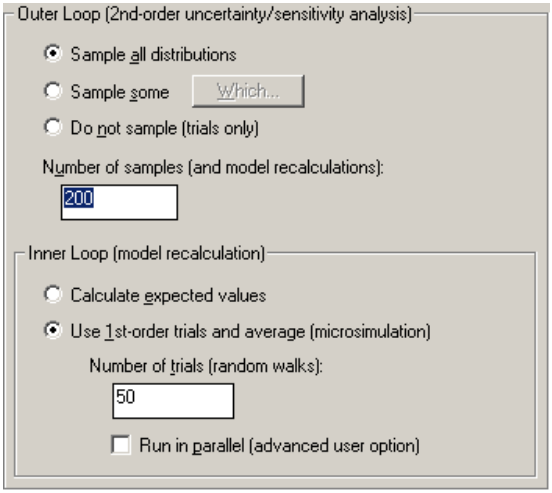
The example model Markov Monte Carlo #2, shown below, uses tracker variables in Markov calculations, which means that simulation trials must be used to correctly analyze the model. Since expected value calculations are not valid in the model, probabilistic sensitivity analysis must be used to perform uncertainty analysis (let's say on pCancer, in this case).

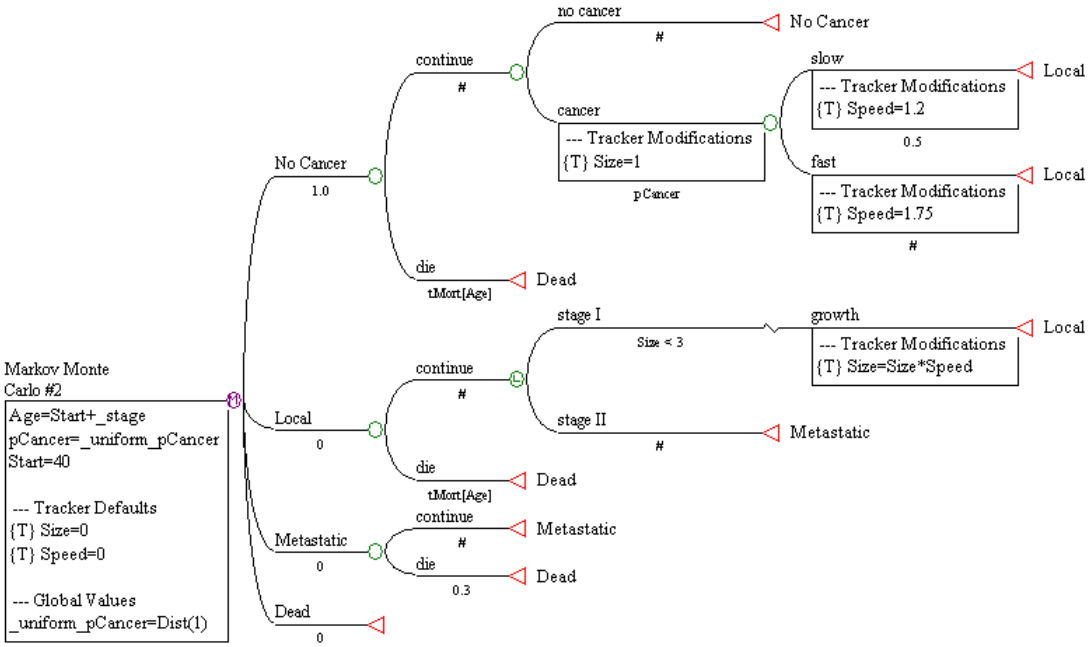
The model requires a probability table called tMort, created in the tutorial in Chapter 35. Create that table now, if you have not already.

► **To perform a probabilistic sensitivity analysis using groups of individual trials:**

▷ In the Markov Monte Carlo #2 tree, select the root, Markov node, and choose **Analysis >** Monte Carlo simulation > Two-Dimensional.

▷ In the Monte Carlo Simulation dialog, specify 200 distribution samples and 50 trials per sample, and then press ENTER of click Begin.





Interpreting output of probabilistic sensitivity analysis using microsimulation

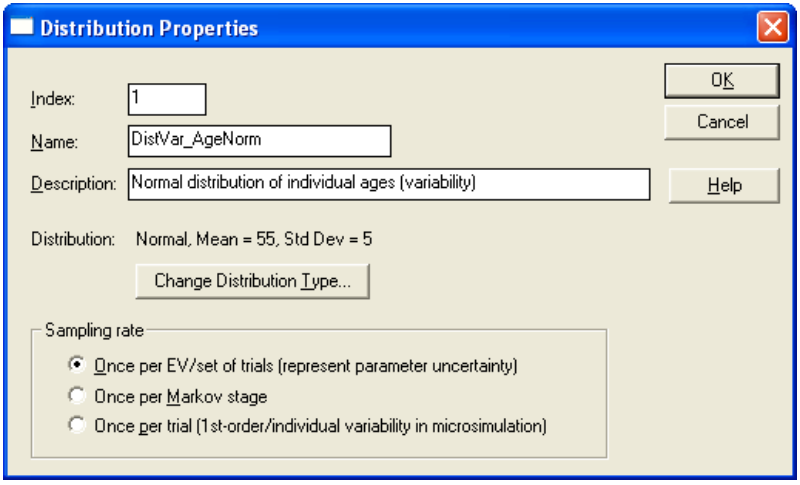
The final output of the simulation will report the statistical summary for the analysis. The statistics are calculated as follows:

- Each group of trials is averaged to produce a mean value which represents that set of distribution samples. This averaging is applied to all output quantities – not just the tree's cost and/or effectiveness attributes, but tracker values and reported distribution sample values, as well.
- The resulting sets of average values are then statistically analyzed.

If you click on the Text Report button in the simulation output window, TreeAge Pro will display a list of averages, each simulating an expected value calculation by averaging the results of multiple trials. For each iteration (i.e., sample), average values are also reported for tracker values and distribution samples, again based on that iteration's group of trials.

Resampling distributions during microsimulation trials

Some distributions are created to represent *variability*, rather than *parameter* uncertainty. These distributions might be used to sample values that vary from individual to individual, or from cycle to cycle. In TreeAge Pro, this is implemented by changing a distribution's sampling rate in the distribution properties. The available sampling rates are: once per group of trials (the default); once per trial; and once per Markov cycle.



See Chapter 16 for information on changing the sampling rate properties for distributions.

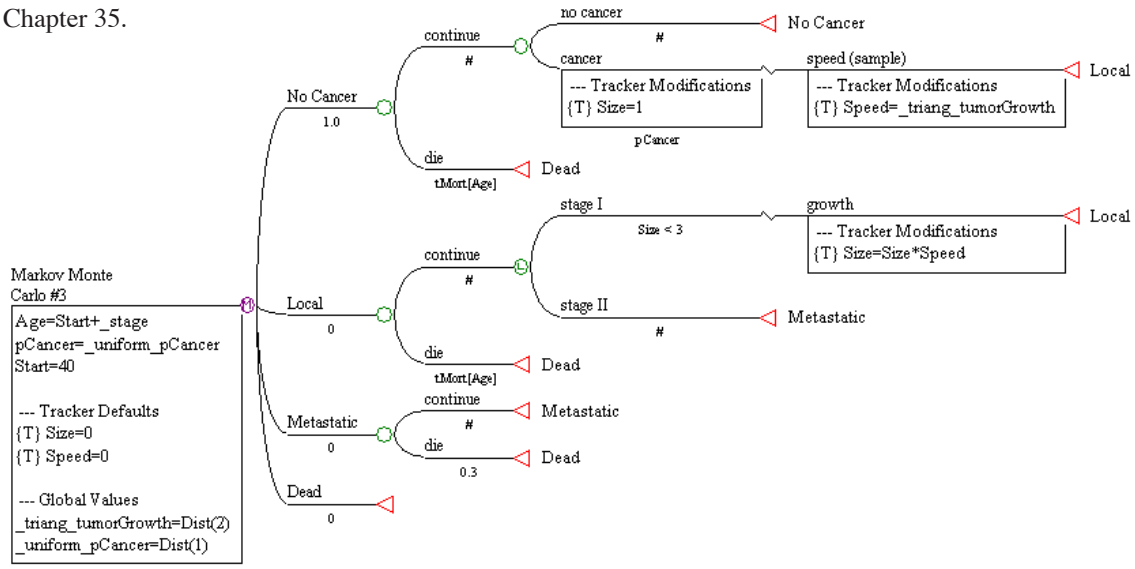
In order for an analysis to include the impact of distributions representing individual variability, a simulation must be run that combines samples and trials. Enabling sampling, in this case, is not necessarily intended as a form of probabilistic sensitivity analysis (although such analysis is still possible on *other* parameters in the model). Instead, its purpose is to include, in each first-order trial, the random effects of sampling the first-order distributions.

The example model Markov Monte Carlo #3, shown on the next page, uses both tracker variables and a distribution that samples the growth rate of a tumor once per microsimulation trial.

Microsimulation must be used to analyze the model; without the tracker variable modifications, no cancer will ever metastasize in the model.

To factor in the variability of the tumor growth rate, first-order distribution samples are used, as well.

The model requires a table, tMort. To try microsimulation on the model, create the table now, if you have not already, following the tutorial in Chapter 35.



Notes on sampling 1st-order distributions

- If distributions are set to sample "per trial" (e.g., because they represent individual or stage-to-stage variability), you can use the **Analysis >** Monte Carlo Simulation > Microsimulation… command to run a one-dimensional microsimulation that also samples from distributions for each trial (similar to running a two-dimensional simulation of N samples x 1 trial each).
- To perform a probabilistic sensitivity analysis on a model which includes both first-order and second-order distributions, the simulation should use two-dimensional simulation, as described in the previous section.
- See Chapter 17 for information on sampling from tables. This approach is used to "bootstrap" patients into the model from a table containing patient data/characteristics (e.g., one row per patient). The built-in keyword **_trial** can be used as a patient counter, to pick the appropriate row predictably, or a uniform (integer-only) distribution can be used to separately sample a patient number for each trial. (In a two-dimensional simulation, the _sample counter increments in the outer loop.)

Sequences and linear sensitivity analysis

In TreeAge Pro, it is possible to create sequences of repeated analyses, including Monte Carlo simulations, as described in Chapter 19. One potential application of the analysis sequencing feature is in creating a linear series of probabilistic sensitivity analyses.

The sequenced simulations approach allows point values to be specified for a parameter and then a microsimulation to be performed, in an automated fashion. This results in something like TreeAge Pro's standard, *n*-way sensitivity analysis.

The analysis sequencing feature first requires that a "template" simulation be created as a stored analysis, to serve as the instructions to TreeAge Pro for each simulation in the sensitivity analysis sequence. The final output of the sequence will not be a sensitivity analysis line graph, but a series of simulation output windows. In order to be able to differentiate between the simulation output windows for the various values of the variable (e.g., pCancer), you will use identifying values in the initial simulation. This way, the value of pCancer will be reported in each simulation output window and its text report.

See Chapter 19 for detailed instructions on storing analyses and sequences of analyses.

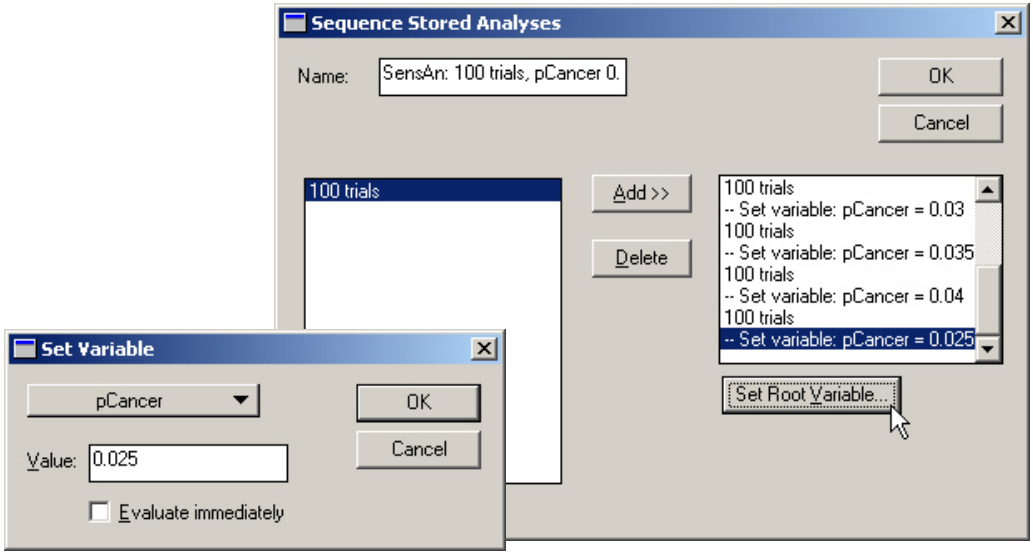
► **To create a sequence of simulations:**

▷ In the Markov Monte Carlo #2 tree, perform a Monte Carlo simulation at the Markov node, specifying Do not sample (in the sequence, you will be manually specifying varying numeric values for pCancer) and 100 trials. Assign pCancer as an identifying value.

▷ Once the simulation is complete, switch back to the tree window and choose **Analysis >** Storage > Save Last… to store the analysis instructions for the Monte Carlo simulation. Enter the name `200 trials` for the stored simulation.

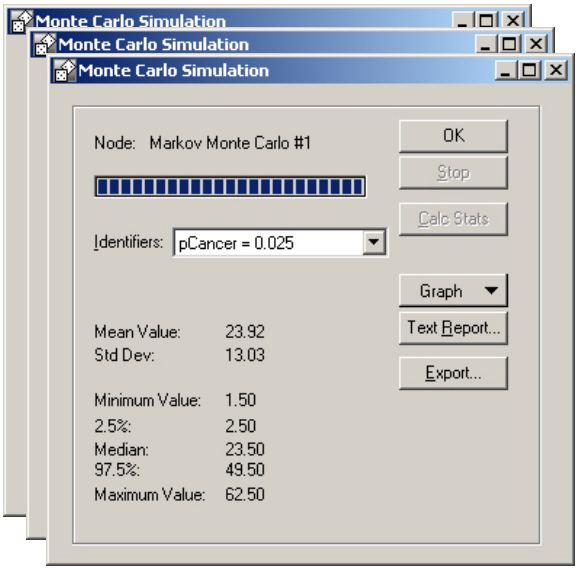
▷ Once you have saved the new stored analysis, choose **Analysis >** Storage > Maintain Analyses…. In the Maintain Analyses dialog, you can now create a *sequence* using the same stored 200 trials simulation many times.

▷ Between each repetition of the simulation, the Set Variable feature is used to modify the value of a variable at the root node.



Once saved, the sequence then becomes a special type of stored analysis, which you (or anyone with TreeAge Pro) can run just like any other stored analysis. To run the linear probabilistic sensitivity analysis sequence you have created, with the tree open, you simply choose **Analysis >** Storage > Run Old Analysis…, choose the sequence from the list, and click the Run button. As described above, the results of this sequence will be a series of Monte Carlo simulation output windows, identified by the value of pCancer for the particular simulation trials.

You can set more than one variable between analyses, if necessary. Also, you can set variables using text formulas instead of numeric values to variables; see Chapter 19 for information on how and when the expressions will be evaluated. For example, instead of setting the variable to a predetermined value at each interval of the analysis, it is possible to assign a random value from a distribution using the Dist() function. Since the variable is being set outside of a Monte Carlo second-order simulation, the Dist() function would normally return the mean value rather than a sample. A special syntax must be used to generate a sample value during expected value calculations, for example p=Dist(*n*;1); see Chapter 16 for details.

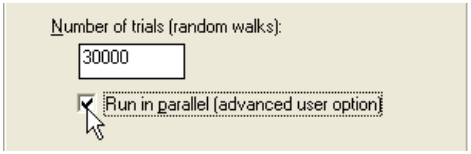


Note:

See Chapter 30 for details on automating analyses using macros and scripts and the Excel/COM module's TreeAgeProLib, instead of using stored analysis sequences.

Running Markov microsimulation trials in parallel

Normally, microsimulation trials happen in *series*. When one trial completes (e.g., a Markov process terminates), the next trial is begun. The TreeAge Pro Healthcare module includes an option to run Markov microsimulation trials in *parallel*. If a Monte Carlo simulation is performed at a Markov node, or at a decision node with Markov branches, the simulation setup dialog presents the parallel trials option below the box where you enter the number of trials to run.



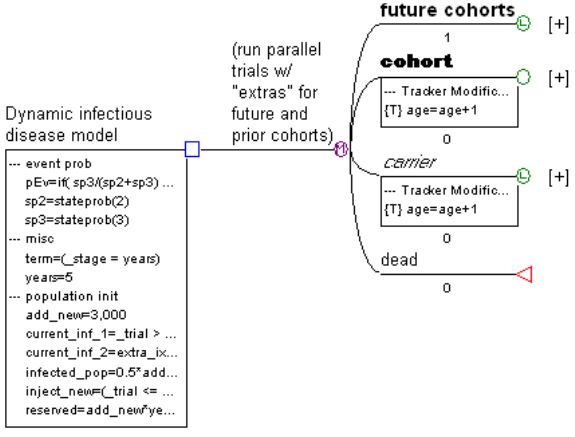
One reason for running trials in parallel is to have access to TreeAge Pro's StateProb() function during microsimulation; it will not work during regular trials run in series. In parallel trials, the "cohort" of *n* trials is stepped through the process one cycle at a time, rather than running each trial to completion before starting another. This allows a variety of kinds of interaction between "individuals," including using the StateProb() function to check how many (really, what percentage of) individuals are currently in particular states, or using the Global functions to enable "communication" between trials/individuals (e.g., queueing, event timing).

An example using parallel Markov microsimulation trials to model an infectious disease in a dynamic population ("Complex Parallel Trials.tre") can be found in TreeAge Pro's Tutorial Examples\ Healthcare subdirectory. The model, shown at right, uses sets of trials/individuals to represent current and future cohorts of interest, injecting a new 15-year-old cohort each year. Another set of trials represents an initial, age-distributed pool of currently infected individuals. An excess of trials is run to ensure sufficient trials for all cohorts.

Another example using parallel Markov trials to model a treatment queue ("Queue Problem.pkg") can be found in TreeAge Pro's Tutorial Examples\ Healthcare subdirectory. The model uses a table (hypothetical patient risk scores) which has been saved with the tree in a package file.

This model has a separate cycle counter — in a tracker variable — because it takes numerous Markov "stages" to fill up the queue for a single treatment cycle with the highest scoring individuals. It takes one stage to record the highest risk score among individuals in a waiting state, then another stage to go back and find the patient with that score and transition them to the treatment queue, and then more stages to repeat this process until the queue is full, and then another stage to actually increment the cycle counter.

IMPORTANT NOTE! One model adjustment that must be made when using parallel trials is that if 1st-order distribution samples are used, their values must be saved into tracker variables (or a Global matrix) during the first cycle (stage 0). Otherwise, the last trial's sampled value is used (as seen in the microsimulation text report). This technique is used for the sampled age in the Complex Parallel Trials example.

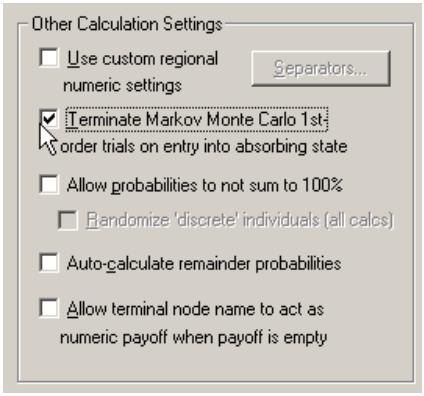


Other aspects of microsimulation

Markov microsimulation is a very broad topic, and TreeAge Pro includes numerous microsimulation options and features which may be of interest in select instances.

Markov termination during simulation trials

By default, Markov process calculations terminate during microsimulation trials when *either* the termination condition is true *or* a trial enters an absorbing state. It is possible to turn off the latter behavior, so that trials will continue processing even after entry into an absorbing state; changing this default setting is likely to be appropriate in only a very few Markov models (e.g., where absorbing states do not correspond to dead states or other endings of the Markov process). This setting is found in the Other Calculation Settings category of the Preferences dialog.


Other Calculation Settings
Use custom regional numeric settings
Separators...
Terminate Markov Monte Carlo 1st-order trials on entry into absorbing state
Allow probabilities to not sum to 100%
Randomize 'discrete' individuals (all calcs)
Auto-calculate remainder probabilities
Allow terminal node name to act as numeric payoff when payoff is empty

The Terminate upon entry into absorbing state setting generally substitutes for the threshold portion of the default Markov termination condition (i.e., "_stage_eff < 0.001"). The threshold part of the condition should generally be removed for microsimulation models.

Using logic nodes in a Markov microsimulation

A *logic node* acts like a decision node, in that it selects one path from its branches; rather than looking at expected values, however, it chooses a path by evaluating logical expressions. Starting at the top branch, the first node with an expression that evaluates to *true* is selected. A simple logic node might have two branches, X and Y, with the expression _stage > 4 below branch X and _stage <=4 (or #) below branch Y. When the logic node is encountered, either branch X or branch Y is followed, based on the current value of _stage.

Most logical expressions in Markov processes will reference the values of tracker variables. For example, in Markov Monte Carlo #2 and #3, a tracker variable serves to remember the current size of a tumor during a simulation trial, while a logic node determines whether to transition to a Metastatic state based on the value of this tracker.

Chapter 37

Markov Technical Details

This chapter covers some technical details related to Markov models in TreeAge Pro.

In this chapter:

- Changing the storage location of tables 518
- Markov cohort EV calculation details 518
- Markov microsimulation trial details 519

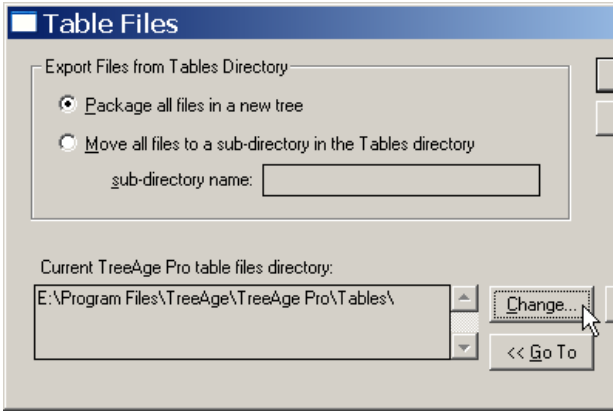
Changing the storage location for tables

The default location of the global Tables directory is in the TreeAge Pro application directory; however, it is possible to change this directory. This is useful for network installations, for example, where the software is stored on a network drive on which different users' tables may conflict. There are two ways to accomplish this change.

Chapter 18 described setting the Tables directory location via a program preference.

► **To change the tables directory:**

▷ In the Variables and Tables dialog, click the **Backup/Move…** button to open the Table Files dialog.


Table Files
Export Files from Tables Directory
Package all files in a new tree
Move all files to a sub-directory in the Tables directory
sub-directory name:
Current TreeAge Pro table files directory:
E:\Program Files\TreeAge\TreeAge Pro\Tables\
Change...
<< Go To

▷ Click the Change… button and provide a new location for tables.

Another way to change the Tables directory, you can create a text file in the same location as TreeAge Pro, and in it specify the name of the directory to be used for table storage.

When you start TreeAge Pro, it checks in its directory for a plain text file called "table.dir"). The first line of the file should be the path of the directory which will be used for table storage. Make sure to add a carriage return (press Enter) after the path.

For example, your table.dir file might contain the text "C:\Project X\Models\Tables" to cause TreeAge Pro to look in that directory for tables. If the specified directory does not exist, TreeAge Pro will ask if you want to create the directory.

TreeAge Pro re-reads the contents of the tables directory frequently, so it is possible to copy tables you need into the specified directory at any time. Any changes you make to the table.dir file, however, will only be recognized by TreeAge Pro the next time you start the program.

It is also possible to specify a tables directory on the command line, using a shortcut to the TreeAge Pro executable.

Markov cohort EV calculation details

Listed below are the steps which TreeAge Pro takes, in the order they occur, during a Markov cohort expected value analysis.

Initialization phase

- The values of _stage, _stage_reward, and _total_reward are set to zero.
- The cohort is distributed among the states according to the initial probabilities, and initial rewards are accumulated.
- Tallied initial rewards are added to _stage_ reward and _total_reward.
- The cohort is redistributed according to the transition probabilities, and transition rewards are accumulated.
- Tallied transition rewards are added to _stage_ reward and _total_reward.

Processing (incremental) phase

- The value of _stage (and, where appropriate, _tunnel) is incremented.
- The termination condition is checked. If it is true, the Processing phase is exited.
- Incremental rewards are accumulated and added to _total_reward.
- The cohort is redistributed according to the transition probabilities, and transition rewards are accumulated and added to _total_reward.
- The value of _stage_reward is reset to zero, then tallied incremental and transition rewards are added to _stage_reward.

Final phase

- The value of _stage_reward is reset to zero.
- Final rewards are accumulated and added to _stage_reward and _total_reward.
- The value of _total_reward becomes the value of the Markov process.

Markov microsimulation trial details

Listed below are the steps which TreeAge Pro takes in evaluating a Markov model during a microsimulation trial.

Monte Carlo trial initialization

- The values of _stage, _stage_reward, and _total_reward are set to zero.
- Per-stage distributions are sampled, if any.
- The trial chooses a state in which to begin the process according to the distribution of initial probabilities, and that state's initial reward is accumulated and added to _stage_reward and _total_reward.
- The trial chooses a new state according to the distribution of transition probabilities, and any transition rewards that are accumulated are added to _stage_reward and _total_reward.

Monte Carlo trial processing phase

- The value of _stage (and, where appropriate, _tunnel) is incremented.
- Per-stage distributions are resampled.
- The termination condition is checked. If the option Terminate Markov Monte Carlo simulations on entry into absorbing state is turned on (in the Other Calc Prefs page of the Preferences dialog), that particular condition is checked first. If either is true, the Processing phase is exited.
- The current state's incremental reward is accumulated and added to _total_reward.
- The trial chooses a new state according to the distribution of transition probabilities, and any transition rewards that are accumulated are added to _total_reward.
- The value of _stage_reward is reset to zero, then the tallied incremental reward and transition rewards are added to _stage_reward.

Monte Carlo final phase

- The value of _stage_reward is reset to zero
- The current state's final reward is accumulated and added to _stage_reward and _total_ reward.
- The value of _total_reward becomes the value of the Markov process for this trial.

Tracker modification timing

Tracker modifications at a node occur after rewards (state or transition) are accumulated at that node. Tracker modifications at a node are evaluated in reverse alphabetical order (from "Z" to "_").

APPENDICES

- Appendix A, "Changes from Earlier Versions"
- Appendix B, "The Tool Bar, Status Bar, and Menus"
- Appendix C, "Preferences"
- Appendix D, "Operators and Functions"
- Appendix E, "Creating Scripts and Interfaces"
- Appendix F, "Technical Notes"

Appendix A

Changes from Earlier Versions

This chapter provides users of earlier versions of TreeAge Pro a detailed listing of the feature changes and additions in TreeAge Pro 2007.

Users upgrading from recent versions of DATA can also find a list of changes since DATA 3.5.

In this appendix:

- What's new in TreeAge Pro 2007 524
- Changes in v2006 528
- Changes from DATA to v2004/v2005 531

What's new in TreeAge Pro 2007

Below is a list of the notable changes from the original release of TreeAge Pro 2006 to the new TreeAge Pro 2007.

Following this section is the list of changes from DATA up to TreeAge Pro 2006.

What's new in the v2007 Core module

Note that some features listed below were added in Release 1.x of TreeAge Pro 2006.

Documents View – The TreeAge Pro window includes a floating window showing a "tree" depicting all open documents, as well as other active windows open for editing variables, tables, and other formulas/expressions. Clicking on a "node" in the Documents View will bring up the selected window. The Documents View can be resized, and closed/reopened (from the Window menu).

Formula Editor – Actions menu added with a Font... button that allows the user to change the default font for variable definition and formula editing windows. For example, this could be used to display text using an easier-to-read size. Also, the Formula Editor now works to edit probabilities for multiple selected nodes, as well as non-calculation fields (e.g., node labels and text boxes).

Define Variable notes – The Define Variable window allows annotation to be added describing the function/syntax of a particular definition. Also, an **Actions** menu added with a Font... button, Find option, and more.

Variables Report – The Values > Reports > Variables... command now includes new options. For example, node-specific reporting can be done on variables in the tree using options for filtering out undefined variables in the currently selected path.

Node Outline pane – The tree window adds a new dockable, hideable pane which displays the currently selected node in text/outline format. (The syntax for the node outline is the same as the tree outline available in previous versions.) The text outline of the selected node can be modified, and then the node updated with the click of a button. To see examples of outline syntax, simply open a tree and display the Node Outline pane, from the Display menu. The Node Outline pane also has its own new button on the TreeAge Pro tool bar.

User-defined functions – Custom functions can be quickly written in the straightforward but powerful Python language. These user-defined Python functions can access tree variables and functions, Python's powerful built-in functions and objects, as well as the vast array of custom and third-party Python scripts and modules. (See: http://docs.python.org and http://www.treeage.com/support/Python.html)

New functions: (see Chapter 21)

- *User("VarName"; arg1; ...)* is used for calling a user-defined function (see above) when the function requires arguments.
- *Command("Excel"; "Open"; BookName)* is used to open a spreadsheet.
- *Command("Excel"; "Macro"; MacroName; arg1)* is used to call spreadsheet macros (with an optional argument).
- *Command("Excel"; "ExportGlobalMatrix"; BookName; SheetName; CellName; ReportLabel)* is used to export the Global( ) function's matrix to Excel. Book, sheet, cell names and label are optional.

Variables display – The "wrap" option has been improved to avoid expanding the width of nodes whose variable definitions are short enough not to require wrapping. Also, user-defined Python function definitions (new in v2007) will be displayed completely if the "wrap" option is turned on.

Tables – The global table files directory can be changed within TreeAge Pro. Go to Variables and Tables dialog, and click on the Backup/Move button. Changing the table files location here sets a preference which overrides any "table.dir" text file settings. Use the Reset button to revert (deletes the preference setting).

Docked Finder – The Options > Find/Replace tool can now be opened as a dockable window (under the Display menu). The regular Find/Replace tool window is still available, too.

Statistics – Statistical reporting runs much faster on very long lists of numbers (i.e., hundreds of thousands of simulation output rows).

Simulations – Virtual memory requirements for storing the running output during very long simulations (or on trees with many outputs) are handled better. Under Windows XP, for example, running more than 1 million iterations on a tree with 150 output columns (e.g., C/E for 5 strategies with ~25 trackers) previously would exceed virtual memory allowed to a program (~1.5GB). A new, user-specified threshold memory size setting (default=500MB) is used to turn on swapping of outputs to temporary files. Note that this requires sufficient space on TEMP folder drive.

Debug pane – A new preference to limit lines of output in debug pane will speed up its operation, especially during simulations. Also, there is an option to turn on/off wrapping of long lines of text. The window includes an ACTIONS menu button with Find, Save, Copy, Font, and other useful commands.

Distributions – Option to approximate parameters for Beta, Gamma, and Lognormal distributions are easier to use.

Influence diagrams – New preferences allow user to control the style of border around either note boxes or arc information notes. Also, arcs between incomplete nodes (e.g., no alternatives) can be displayed as gray dashed lines, like structure-only arcs.

Variable categories – New options are available for formatting or hiding separators between variable categories displayed in the tree window. [v2006 release 1]

Custom payoff names – Using a new calculation method preference, custom names can be assigned to the nine payoff sets in a tree. The custom payoff names are used to label payoffs and rewards in the Properties Grid, Enter Payoff, Markov Rewards, and Multi-Attribute Weights windows. [v2006 release 1]

Properties grid – If node comments are entered for the branches of a parent node using OPTIONS > Node Comment…, the comments will appear (and be editable) in the Properties grid. Also, the Properties grid handles EDIT > Copy/Paste Text commands, making it easier to edit formulas and text. [v2006 release 1]

Roll back columns – Using a new tree preference, custom text can be displayed in a notebox above roll back columns, if they are active. [v2006 release 1]

Monte Carlo scatterplot – Holding down the CTRL key while moving the mouse over either the C/E or incremental C/E scatterplot graph area will report the number of simulation results/points within a few pixels of the mouse pointer. This is useful, for example, when trying to determine whether many identical results are stacked on the same point (e.g., the origin in the ICE scatterplot, where two strategies are equal – see below). [v2006 release 1]

Monte Carlo seeding options – The "Independent" seed option works during simulations on multi-processor computers. Also, a fourth seeding option is added ("Start microsimulation loops with same seed...") which allows for variation/error reduction during 2- and 3-dimensional simulations (e.g., probabilistic sensitivity analysis or partial EVPI) which use microsimulation rather than EV calculations. (In other seeding options, the position in the random number sequence changes from one set of trials to the next, consistent only between decision strategies in a single trial). [v2006 release 1]

Tracker variables – Added a preference to enable tracker modifications for expected value calculations in regular trees. (Only works from right to left, in non-Markov tree calculations; tracker values will not be remembered from previous cycles in expected value Markov cohort calculations.) [v2006 release 1]

Clones – Clone update suppression can be maintained during Markov Monte Carlo microsimulation; previously not enabled (also see memory-intensive tree preferences, below). Improvements have also been made to speed up clone copy updating. [v2006 release 1]

Memory-intensive trees – Large tree optimization preferences are collected in a new Preferences category, including a new preference for turning off variable calculation/search optimization in extremely large, memory-intensive models. [v2006 release 1]

Text documents – Text document window includes Find command for simple text searches. [v2006 release 1]

What's new in the v2007 Excel/COM module

Import/Export values to Excel – The updated Excel module also includes changes to the Export Variables (from TreeAge) command as well as the Excel add-in's commands for adding/updating variables from the spreadsheet. Exporting a variable to Excel now includes extra information, allowing variables to be renamed from the spreadsheet. Also, when a non-root node is selected in the tree, exporting a variable to Excel includes its node-specific definition (as well as a default definition, if any). New TreeAge menu commands in Excel enable worksheets already including variables and tables to be "refreshed," in order to retrieve newer/modified versions from the tree.

Scripting/automation – TreeAge Pro Suite and TreeAge Pro Excel both now support a full-featured scripting interface, very similar to the TreeAge Pro Interactive interfaces. The TreeAge Pro 2007 Excel/COM automation library (called TreeAgeProLib) provides macros/scripts with access to a variety of modeling and analysis features in the TreeAge Pro application.
(NOTE: The old DataATLLib.CurrentTree object is no longer available, its functions now handled by TreeObj and ApplicationObj objects in the TreeAgeProLib.)

Excel add-in – The Excel module's TreeAge menu shows the name of the current tree, and can convert a text outline to tree (without creating/opening a text file in TreeAge Pro). Exporting values, reports and graphs provides an option to utilize an existing worksheet in the current workbook if it has the correct name.
[v2006 release 1]

What's new in the v2007 Healthcare module

Net Benefit/CEA – In cost-effectiveness models, 2-way & 3-way sensitivity analysis and threshold analysis now can use Net Benefits calculations, making cost-effectiveness thresholds easy to identify. Also, a new Net Benefits command under the Analysis menu generates a line graph showing how each strategy's Net Benefits change as the willingness-to-pay increases.

Monte Carlo EVPI – Changes to cost-effectiveness simulation EVPI output include: a curve showing the sensitivity of EVPI (or partial EVPI) to a range of willingness-to-pay values; text reports can be reported using either the Net Health Benefit or Net Monetary Benefit scales.
[v2006 release 1]

Monte Carlo ICE scatterplot report – An indifference component can be created, reporting as "indifferent" points with IC and IE equal to 0 (or within +/- specified tolerances).
[v2006 release 1]

Markov details report – A new space-saving option in the Markov details report allows event tree reporting to be turned off completely (still displays initial and final data for each state and stage). Also, if the "Calculate extra payoffs" tree preference is used, columns are displayed for extra reward sets.
[v2006 release 1]

Markov termination – The VALUES > Markov Termination… command can be used with multiple nodes selected.
[v2006 release 1]

Changes in v2006

Below is a list of the notable changes from v2005 to v2006.

Following this section is the list of changes from DATA up to TreeAge Pro 2005.

What's new in the v2006 Core module

Note that some of these features were added in Release 1.x of TreeAge Pro 2005 (marked "New in v2005 rel. 1.x").

Tables/databases – Tables can be populated and dynamically updated by querying any ODBC data source (including SQL Server, Excel, text files, and more). Also: interpolation works between columns, not just rows, allowing for easier sensitivity analysis on values in tables; table window has a tool bar, and acts like a document, prompting to save changes; sample from multi-column tables using Dist(n; col) syntax. See Chapter 17.

Influence diagrams – Property grid added for quick editing of node names and outcomes/alternatives.

Sequenced analyses – The Analysis > Storage > Maintain Analyses... dialog now has options for editing existing sequences (e.g., to change/remove steps). Sequence steps can be copied as text into a text editor, and then pasted back into the sequence. The ability to add Run Script steps in an analysis sequence enables, for example, Monte Carlo simulation output windows to be saved and closed automatically prior to running the next simulation. See Chapter 18.

Extra panes – The explorer view, properties grid, and other extra tree panes store their display settings as regular preferences with the tree, as well as defaults for new trees.

Variables editing – The Variable Definitions window is now a dockable pane, displayed by default to the right of the tree view; an Actions menu allows defining existing or new variables, as well as copying and pasting lists of definitions. The Insert Variable dialog separates lists of variables, tables, and named distributions.

Roll back columns – Improvements to setup dialog and column formatting. See Chapter 10.

Monte Carlo options – Special settings such as seed value and type, number of threads, and "sample some" distributions can be saved with tree. New "independent" seeding option makes it possible to avoid changes in all results of a seeded simulation when only changing the number of sampling distributions or decision node branches. See Chapter 16. (Also, see new Seed(*n*) function, below.)

Tree window – Cues for beginning users to show where branch probabilities and labels are entered. (Turn cues on/off under Help menu). When editing a branch probability, an ellipsis ("...") button is shown that opens a Formula Editor dialog.

Branching – The Options > Reorder Branches dialog can be used to insert specific numbers of new branches. Control + 0 (zero) will jump to the parent node and collapse it. (Control + 1 through 9 still move selection to corresponding branch and uncollapse it.)

New functions:

- *ProbToProb(p;t)* is shorthand for "RateToProb(ProbToRate(p;1)*t; 1)".
- *BranchProb()* returns the calculated branch probability of the node being calculated.
- *BilinkName("range")* can be used instead of Bilink(n) to calculate a dynamic link's value.
- *DistForce(n)* offers the same functionality as the special Dist(*n*;1) function syntax for forcing sampling during non-sampling analyses, but works with multi-variate distributions.
- *Seed(n)* can be used to override the normal random number generator seeding, for example when using the Node() function to run trials.
- The *Command()* function can be used to perform actions on selected nodes ("branchfromtext") or tables ("requery", "save", "clear").
- *Interpolate(x;oldLoX;oldHiX;newLo;newHi)* to rescale a value. See Appendix D.

Packaging tables/trees – If a table in used in a formula, TreeAge suggests package (*.pkg) in the Save As dialog.

Copy Special – Copying picture of subtree to clipboard only requires selecting subtree's root.

Node comments – Included in converted tree outline format. See Appendix F.

Variables grouping – User-defined categories can be used to organize variables displayed in the properties grid and tree window. See Chapter 13. (New in v2005 release 1.x)

Simulation outputs – Monte Carlo simulations now report supporting details on expected value of perfect information (EVPI) calculations, or partial EVPI. Also: a single statistics report is available for all output quantities (including Net Benefits in cost-effectiveness models); completed simulation results are always reported, even if an error causes the simulation to end; graphs and reports can be generated more quickly when many thousands of rows/items exist. See Chapter 15 and the Help file for details. (New in v2005 release 1.x)

Debug pane – Debugging preferences can be set to display tree errors, internal calculations, and other messages in a Debug pane/window (similar to the explorer view and properties grid panes). Possible to step through node-by-node and calculation-by-calculation. Debug() function dynamically controls debugging. See the Help file for complete details. (New in v2005 release 1.x)

Headers/footers – Headers and footers in graphs and influence diagrams. (New in v2005 release 1.x)

Roll back display – New options for displaying calculated branch and path probabilities (e.g., scientific notation) are available in the tree Numeric Formatting dialog. Also, a new option in Roll Back preferences can be used to display both path probabilities and expected values at non-end nodes. (New in v2005 release 1.x)

Node numbering – A new sub-option in the Terminal Node preferences enables the node numbering scheme to be applied to all tree nodes, not just terminal nodes. (New in v2005 release 1.x)

Variable/formula editor – To better handle complex definitions, windows are resizable, and functions can be displayed automatically using indenting. (New in v2005 release 1.x)

Distributions – Includes a visual tool for building custom table distributions based on a built-in distribution. Some distributions (log-normal, beta) can now be approximated from common statistics. Distribution reports, including sample statistics, have been added. See Chapter 16. (New in v2005 release 1.x)

Python functions – Embedded Python script interpreter (v2.4 installed; open source, freely distributable; info: http://www.python.org). A new set of functions allows TreeAge Pro to use Python modules or the Python script interpreter for random number generation, matrix manipulation, scientific programming, etc. Or, you can write your own Python scripts/modules. See Appendix D. (New in v2005 release 1.x)

Tree() function – Dynamic linking between a master tree and any number of hidden trees can be set up using the Tree() function. See Appendix D. (New in v2005 release 1.x)

Node() function – Can calculate and cache statistics other than the mean for microsimulation trials run at the target node. See Appendix D. (New in v2005 release 1.x)

Script() function – Run TreeAge Script commands on visible or hidden trees. See Appendix D. (New in v2005 release 1.x)

Reports – Text reports for the lists of distributions and dynamic links. (New in v2005 release 1.x)

Arrows – A new option in the Notes & Arrows preferences for trees enables arrow tails and heads to be anchored to nodes, so arrows move as the tree layout changes.

Check for Updates – TreeAge Pro can automatically check for updates. (New in v2005 release 1.x)

Changes in the v2006 Excel module

Distribution editing – New Edit in Excel option available in the Distributions list (and add-in menu in Excel includes corresponding Add/Update Distributions... command).

Exporting reports – When exporting reports (and distribution graphs) which exceed Excel's worksheet size of approximately 65,000 rows, TreeAge Pro prompts for a range of result rows to include.

Changes in the v2006 Healthcare module

Markov bindings – A new dockable editing window enables easy editing/copying/pasting of Markov state bindings at a node, including from other trees or a text editor/spreadsheet.

Markov speed-up – 25% to 800% microsimulation speed improvement for some Markov models, depending on the model (e.g., number of tracker variables) and the type of simulation. Also, expected value analyses other than the Full Detail Cohort Analysis use less memory, so extremely complex models may run faster. (New in v2005 release 1.x)

Changes from DATA to v2004/v2005

Below is a list of the notable differences from DATA 3.5 to TreeAge Pro 2005.

Core module additions/changes

Properties grid – Each tree includes a *properties grid*, displaying and allowing editing of a wide variety of information at the currently selected node, including variables, probabilities, payoffs, and Markov information. Tables can also be edited in the properties grid (if they are not very large).

Explorer and properties grid docking – The explorer view and properties grid panes can be independently docked on any edge of the tree window, or floated in or outside the tree window, for easier debugging of complex models.

Packaged trees – When a package is opened, its tables remain in the package with the tree; changes to either the tree or its tables are saved back into the package by default. During analysis of a packaged tree, tables stored in the package take precedence over table files.

Variables and Tables – To enable management of packaged tables and table files, the Variables and Tables dialog separates them into two lists. Options have been added for importing tables and variables, and for backing up tables.

Influence diagram distributions – Sampling distributions can be used in probabilities or values, and will be included in the converted tree.

Influence diagram quick menus – More frequently-used commands have been added to the right-click menus.

Imports – Using their respective dialogs, it is possible to import a list of distributions, a set of preferences, or a list of variables into the current tree from another saved tree.

Distribution preview – Most distributions can be previewed from the Distributions dialog list, using the Graph It button to generate a histogram.

Distributions statistics – The formatting of the statistics report can be set independently of the histogram. Also, the text report can be exported.

Distribution graph axis scaling – The vertical, probability axis in distribution graphs is automatically scaled, taking into account the height of the highest current bar, when the graph is created or its horizontal axis is modified.

Tool bar – The tool bar in the tree window now includes buttons for two existing tools: the Calculator/Evaluator window and the Formula Editor window. Both are now accessible from any value editor.

Text report to text file – Text reports can also be "exported" to a new text document window within TreeAge Pro.

Backup/Autosave — Automatic backup and autosave capabilities have been added to TreeAge Pro. A new category page has been added to the Preferences dialog to handle backup and autosave options. (New in release 2004.0.7)

Text file format – TreeAge Pro includes a general usage text file format for making simple edits to table data, reviewing text report results, creating a rollback text report, and creating tree outlines. (New in release 2004.0.7)

Rollback report – It is possible to view a textual representation of a rolled-back tree. (New in release 2004.0.7)

Tree outlines – A tree outline is a simple text representation of a decision tree which can be converted into a TreeAge Pro tree file. The text outline can be created in a spreadsheet or word processing program, saved as a text document, opened in TreeAge Pro, and then converted into a tree. (New in release 2004.0.7)

Graph markers – Graphs that use marker symbols, such as line graphs and cost-effectiveness graphs, now include English alphabet characters (uppercase only) in the list of available symbols. See Chapter 6. (New in release 2004.0.7)

Window size – TreeAge Pro will automatically resize a maximized tree window when a value editing window is displayed for the tree, so that the tree will be partly visible. When you close the editing window, TreeAge Pro will automatically re-maximize the tree window. (New in release 2004.0.7)

Status bar – When a node is selected, the status bar reports information about the node in the status bar: currently-selected node type, branch count, and variable definition count. See Appendix B. (New in release 2004.0.7)

Influence diagrams/Bayes' revision – When converting an influence diagram to a tree with numerous, sequential Bayes' revision event nodes, TreeAge Pro can efficiently create thousands of variables, if necessary, for probability revision. (New in release 2004.0.7)

Suppressable dialogs – Selected user prompts and message boxes have been made suppressable. In these cases, a checkbox is provided that instructs TreeAge Pro not to show the dialog again. If a dialog that asks a question is suppressed, TreeAge Pro will store and continue to use the last answer provided, where appropriate. All suppressed dialogs can be re-enabled via the **Help >** Enable Suppressed Dialogs command. (New in release 2004.0.7)

Tree preferences – When opening a tree (or its Preferences dialog after open), holding down the CONTROL key will change the tree's preferences back to your current defaults. Holding down the SHIFT key while opening the Preferences dialog will, instead, delete the current default settings and return to the program's initial defaults. (New in release 2004.0.7)

Loading table data – A Load Text File… command makes it easy to import large, tab-delimited tables when copying/pasting is difficult. In addition, a new Save Text File… command can be used to export the table data to a tab-delimited text file. (New in release 2004.0.7)

Risk preferences – When entering a non-constant risk preference function, a wider payoff range is used.

Dynamic linking to Excel – The Edit menu now includes a Links sub-menu, with a new Dynamic Links command. Dynamic links and the Bilink() function are designed to replace both bi-directional and DDE links, and are the default format for new links. Dynamic links are more reliable, faster, and easier to setup and maintain than old DDE and bi-directional links.

Simulation duration – The estimated time remaining in a simulation is reported.

Link to cell ranges – Any single link (dynamic or otherwise) can be to a range of spreadsheet cells.

Multi-level Monte Carlo simulation – TreeAge Pro employs a more flexible simulation algorithm. It includes options for two- and three-level simulations. Multi-level simulations are useful, for example, in *value of information* analysis. A more sophisticated random number generator supports simulations of up to 5 million iterations.

Simulation output – The simulation output window itself can be saved and reopened or shared with other users. TreeAge Pro provides separate graphs and reports for each strategy at a decision node.

Simulation using multiple processors – To support complex, lengthy Monte Carlo simulations, TreeAge Pro can use up to 8 processors on a single computer.

Tree Explorer view – The tree window now includes a compact tree view side by side with the regular tree view. The Explorer view makes it easy

to navigate in very large models, and includes cut/copy/paste subtree functionality. See Chapter 21.

"Skeleton" tree display – It is possible to display a "skeleton" view of a tree by turning on a tree preference. This will display subtrees as floating slightly to the right of the parent node, and collapsed subtrees will be completely hidden, creating a highly compact, abstract view of the tree. See Appendix C.

Wrap long variable definitions – A display preference allows long variable definitions to be wrapped to a fixed multiple of the minimum standard branch width (8/10"). See Appendix C.

Cut/copy/paste definitions – Selected variable definitions can be cut, copied, or pasted at a selected node, via the Variable Definitions window. See Chapter 10.

Enter up to 9 sets of payoffs – Up to 9 sets of payoffs (or Markov rewards in the Healthcare module) can be assigned at terminal nodes (increased from 4 sets in previous versions). See Chapter 7.

Additive/multiplicative chance nodes – TreeAge Pro includes a preference to allow non-coherent probabilities (sum to greater than 1.0, or even less than 1.0). In a regular decision tree, this can be used to have chance nodes in which the branches are additive rather than mutually exclusive. (In the healthcare module, this means that a Markov model can perform calculations based on a population of individuals, and even change the size of the cohort over time.) See Chapter 22.

Incoherent probabilities/discrete "populations" –With non-coherent probabilities enabled in the calculation preferences, TreeAge Pro can perform expected value calculations using non-coherent probabilities. However, an additional preference can be used to randomize "individuals" at chance nodes based on probabilities, always keeping them "whole." See Chapter 22.

PathProb() function – This new function returns the cumulative path probability calculated for the node being calculated (the Markov node in a Markov subtree). See Appendix D. Users of the Healthcare module also have a new **StateProb()** function; see the Healthcare module documentation for details. See Appendix D.

MatrixMult(), Matrix(), Global() functions – MatrixMult() performs matrix multiplication on two matrices, returning a cell from the resulting matrix. MatrixMult() can be used to correlate a list of distribution samples using a covariance matrix; use Matrix() to input matrices, with a specified number of values per row. The Global() function is a dynamic table/matrix which can be used to save custom reports during complex analyses. See Appendix D.

New distribution functions – TreeAge Pro includes a variety of new functions related to sampling distributions. The *DistKids()* function dynamically creates (hidden) branches at a chance node, including internal chance nodes, during an analysis. The *DistTrim()* function truncates distribution samples by forcing resampling until a number in the desired range is returned. The *DistProb()* function returns the cumulative probability of a particular value in a selected sampling distribution. The *DistValue()* function returns the value of a selected sampling distribution for a particular cumulative probability. See Appendix D.

10/50/90 fractile distributions – TreeAge Pro's palette of distributions includes new options for creating discrete distributions based on 3 fractiles/percentiles. The available forms are: .10, .50, .90 fractiles with 30%, 40%, and 30% sampling probabilities; and .10, .50, .90 fractiles with 25%, 50%, and 25% probabilities. See Appendix E.

Dirichlet distribution (multivariate normalized Betas) – The Dirichlet distribution is used to sample probabilities for all branches of a chance node (summing to 1.0). It uses a list of parameters to generate sets of Gamma random variates that are normalized (divided by their sum). See Appendix E.

Uniform distribution, integer form – The Uniform distribution now includes an option to sample only integers in the range specified (each with equal probability). See Appendix E.

Sampling distribution names – In addition to referring to distributions using the Dist(n) syntax, distributions now have a name property, and can be referred to in formulas by name, without having to create a variable. See Chapter 16.

Sampling distribution frequency – In addition to sampling once per 2nd-order iteration, distributions can be set to sample once per 1st-order trial. See Chapter 16.

Sequenced analyses – The analysis storage feature has been extended to enable batch processing of analyses through the use of a new sequencing feature. See Chapter 18.

Line graphs – Multiple graphs can be merged, line colors can be changed, and the coordinates of any selected threshold points on a line can be checked. See Chapter 6.

Regional numeric settings – A number of options are available to facilitate sharing a model between different countries/regions using different numeric settings (i.e., commas vs. periods as decimal and thousands separator characters). See Chapter 7.

Exporting JPEGs – The File > Export dialog includes an option to save pictures of TreeAge Pro files as JPEGs. See Chapter 4.

Clones – It is possible to select "ghost" nodes within clone copies, in order to perform analyses at internal nodes. Trees using clones are stored much more efficiently, so that files are smaller and open/close more quickly. See Chapter 21.

Influence diagram nodes – Double-clicking activates an appropriate node editor. See Chapter 3.

Probability distributions – Selecting a decision node when performing a probability distribution expected value analysis will generate a comparative, cumulative risk profile. It is no longer necessary to multi-select the individual branches of the decision node. See Chapter 5.

Function names – DistSamp() has been shortened to Dist(), UtilDiscount() has been shortened to Discount(). Old names will still work. See Appendix D.

Packaged trees/tables – Trees using tables can be packaged into a single file that can be sent to another TreeAge Pro user. See Chapter 17.

Protected trees – Using the tree packaging process, it is possible to generate a random numeric password for a file. Only a TreeAge Pro user with the correct password can open your package. See Appendix F.

Multi-column tables – Lookup tables (*.tbl files) can have up to 512 columns of values pasted in from a spreadsheet. See Chapter 17.

Tables Used report – This report now can be exported, including the contents (column titles, indexes, and values) and properties of all tables referenced in the tree. See Chapter 17.

Values dialog changes & improvements – For example: The Define Values has been renamed and improved. The new Variables and Tables dialog has resizable columns, options for filtering variables or tables, and links to the variables and tables text reports.

Changes from v2004 to the v2005 Excel module

Here is a list of notable differences between the TreeAge Pro v2005 Excel module and its original release with TreeAge Pro 2004.

Distribution chart statistics – Distribution chart statistics are exported along with the distribution graph or text report.

Distribution chart settings – Distribution chart settings are now saved with the worksheet, making it easy to modify chart settings when reopening the worksheet.

Text report exports – Additional graphs and text reports have the Export to Excel option enabled now.

Tables and Variables – The Variables and Tables dialog in TreeAge Pro includes an **Edit in Excel** button, which can be used to create worksheets in Excel for editing the selected tables or variables. The add-in menu in Excel has a corresponding **TreeAge >** Add or Update Table command; after making changes in a worksheet, it can be used to either update or create tables and variables in the active tree. See Chapters 17 and 29.

Changes from DATA to the Healthcare module

Find/Replace jump states/bindings – The Options > Find/Replace dialog can search for (and replace) text in Markov jump states (at transition nodes) as well as Markov transition binding definitions.

Markov cohort analysis details – New, customizable Markov cohort analysis 'Expanded Report' showing complete detail for all transitions, all states, all cycles. Includes breakdown of multi-attribute costs.

Markov decision processes – Decisions within Markov processes enabled for expected value calculations; cohort analysis and roll back report cycle-by-cycle optimality in Markov decision process.

Cost-effectiveness report options – New cost-effectivness analysis text report options allow comparison of all strategies to a selected baseline, as well as ordering strategies by effectiveness instead of cost.

Markov Information tool bar button – New button activates appropriate Markov information editing dialog at or to the right of a Markov node.

Tunnel states – Markov Tunnel states are more efficient, allowing very large arrays of temporary states. Also, a tunnel's temporary states can be collapsed in Markov cohort analysis reports and graphs.

Tunnel counter/bindings – New special Markov binding names "Tunnel Variable" and "Tunnel Table" can be used to control the _tunnel counter.

Acceptability curves — Generate true net benefit acceptability curves from the results of cost-effectiveness simulations. (Updated from DATA Pro.)

Multi-attribute cost calculation — When performing cost-effectiveness calculations, a weighting function can be used to combine up to 8 payoffs/rewards for the cost component of cost-effectiveness calculations. This facilitates separation of the parts of a complex cost formula (for example, fixed versus variable costs).

Tornado diagrams — TreeAge Pro can generate tornado diagrams reporting either net monetary benefits (NMB) or net health benefits (NHB) in cost-effectiveness models.

Net benefits — TreeAge Pro can generate both net monetary benefits (NMB) and net health benefits (NHB) reports and graphs from Monte Carlo simulation, 1-way sensitivity analysis, and tornado diagrams of cost-effectiveness models.

Inverted effectiveness — In TreeAge Pro, cost-effectiveness trees can use a measure of effectiveness that should be minimized, such as adverse events, rather than maximized.

ICER scatterplot confidence ellipse — In the ICER scatterplot, TreeAge Pro calculates the covariance of incremental costs and incremental effectiveness, and displays this information in the form of confidence ellipses.

Survival curves — TreeAge Pro automatically generates survival probability curves, based on the user's selection of the non-survival (e.g., Dead) state or states.

Tunnel states — Using a Markov state binding named "Tunnel Crossover", transitions between two tunnel states can be made to continue incrementing the same _tunnel counter. A new setting makes it is possible to use the _tunnel counter in the initial probability expression for a tunnel state, in order to populate the temporary states (e.g., "Alive_Tun_2" and so on) prior to transitions (i.e., prior to stage 0).

Finite, dynamic cohorts — It is now possible to create Markov models having a discrete number of individuals in a cohort (without using 1st-order Monte Carlo simulation). Non-coherent transition probabilities can also be used to change the size of the cohort over time (e.g., to handle births or new health plan members).

Find/Replace Markov info — The improved Find/Replace dialog in TreeAge Pro can find and replace text used in Markov termination conditions, state rewards, and transition rewards.

StateProb( ) function — This new function can be used during Markov calculations to reference the percentage (or number) of cohort members that are in one or more states at the beginning of any cycle.

Global( ) function — This new function has many possible uses. For example, it can facilitate more flexible and complex types of Markov modeling by allowing a dynamic matrix to be populated or referenced from any node in the tree (even outside a Markov process). The matrix can also be saved to a text file dynamically, at any point during a calculation.

Node( ) function — This new function can be used to link a formula to the dynamically calculated (or simulated) value of a node elsewhere in the same tree. The Node function can be used in combination with tracker variables and the Global function (see above) to link many terminal node payoff calculations to one Markov node elsewhere in the tree, or to nest Markov models.

Parallel trials — In addition to running a regular series of 1st-order simulation trials for a Markov model, TreeAge Pro offers the ability to run a set of trials in parallel at a Markov node or a decision node with Markov branches. Rather than running one trial at a time, to completion, this option will run the specified number of trials one stage at a time. This facilitates use of the StateProb and Global functions during the 1st-order simulation.

Per-stage sampling — Distributions set to sample every Markov stage will do so during EV/cohort recalculation of the model, not just during 1st-order trials.

Tracker variables display — Tracker variable modifications/definitions are displayed under a separate heading beneath nodes. Their initial values can be displayed at the root node, beneath variable definitions and named distributions.

Appendix B

Tool Bar and Menu Reference

This appendix describes each of the major elements of the TreeAge Pro window: the tool bar, the status bar, and the main menus.

In this appendix:

- The tool bar 538
- The status bar 539
- The menus 540

The tool bar



	Tree Window only	Influence Diagram only
File > New	Values > Show Node Outline Pane	create new arc
File > Open	Values > Show Variable Definitions Window	create new decision node
File > Save	Values > Variables and Tables	create new chance node
File > Print	Values > Markov Info	create new deterministic node
File > Print Preview	Values > Formula Editor	create new value node
File > Convert	Values > Calculator/Evaluator	
	Values > Probability Wheel	
Edit > Preferences	Options > Change Node Type	
Display > Font	Options > Node Comment	
	Options > Custom Node Action	
	Analysis > Probability Distribution	
	Analysis > Sensitivity Analysis > One-Way	
	Analysis > Sensitivity Analysis > Two-Way	
	Analysis > Monte Carlo Simulation >	
	Analysis > Roll Back	
	Navigation button (see below)	

Navigation button

The navigation tool bar button acts as a toggle between two modes of using the arrow keys in the tree window. When navigate mode is active (i.e., the button is down), the arrow keys will change which node is selected. For instance, if a single node is selected, and you press the left arrow key, TreeAge Pro will select the parent of the selected node.

When navigate mode is not active (i.e., the button is up), the arrow keys operate on the text insertion cursor. It is possible to use the arrow keys to maneuver the node selection even when the navigation button is up. This requires holding down the CONTROL key when using the arrow keys.

In all windows other than the tree window, arrow keys operate on the text insertion cursor.

The status bar



1. This area displays information about the currently-selected menu item , tool bar button, or tree node. On occasion, other context-specific information is displayed in this area.

2. When the software being used is the run-time version, this area will display "RUNTIME." The text also displays when the full version is being used with the "Mimic run-time" option, as set in the Options menu.

3. This area displays the text "RISK" when the active tree is set to calculate using a risk preference function, rather than using expected values.

4. The current calculation method for the active tree is displayed in this area. For example, if the tree is set to calculate cost-effectiveness, with payoff 2 used for cost values and payoff 1 used for effectiveness values, this area will display "C/E, 2/1."

The status bar will also display a progress indicator when a long analysis is being run. Because this is such a useful indicator, it is desirable to leave the status bar visible at all times.



File menu

New…



Creates a new document — either a tree or influence diagram. This option allows you to build a new model from scratch.

Open…



Presents the Open file… dialog box for you to open an existing model or graph.

Close

Closes the active window. If changes have been made to the document since the last time it was saved, choosing Close will be followed by a dialog giving you the option of saving, not saving, or canceling the close request.

Save



Saves the document in the currently active window. If a file name has been specified, the document is saved under that name; otherwise you must enter the file name under which it should be saved.

Save As…

Same function as Save, but you are prompted for a file name, whether or not one has already been specified.

Export…

Allows you to export your document as a picture (metafile or bitmap) for use in a word processor or other program. You may also use the Export command to save earlier version models, or a TRB file, which is a permanently rolled-back picture of your tree, viewable only in TreeAge Pro.

Revert to Saved

Reverts to the most recently saved version of the document in the active window. This option should be used if you want to eliminate all changes made to a document subsequent to the last time the document was saved.

Convert



If an influence diagram is in the active window, this command converts it into a tree.

Print Preview…



Displays the document in the active window, indicating the location of page breaks and the number of pages required to print it (in accordance with the options selected under Page Setup). Buttons available in the preview window permit direct access to Page Setup and Print.

Page Setup…

Displays the standard Page Setup dialog box for the particular printer you are using.

Print…



Prints the document in the active window.

Run Script…

Allows you to run a text file containing commands in the TreeAge ProScript language.

(File Names)

TreeAge Pro displays a list of your most recently opened files. Selecting one from the list will reopen that document, so long as its name and location have not changed since the last time it was opened.

Exit

Exits the application, after offering the option to save any changes made to documents open when this command is given.

Undo

This command permits you to reverse (undo) your most recent action if, for example, you issued the wrong command or made a mistake in typing or tree construction. In the case of typing or text formatting, this command will reverse multiple, consecutive changes.

Redo

Re-executes the most recently undone action.

Cut…

Cuts the selected portion of the active window onto the clipboard. This can result in one of several different actions, depending on the context. For example, if a tree window is active, the Cut command will read, depending on the context, “Cut Node” (when no branches emanate from the selected node), “Cut Subtree,” “Cut Variables,” “Cut Note,” or “Cut Text.” When it is not clear whether node text or a portion of the tree is to be affected, the command will read “Cut...” and will be followed by an appropriate dialog box. A node (together with the branch leading to it) or a subtree is cut to the active tree clipboard (see below).

If an influence diagram window is active, you may cut one or more nodes and the arcs connecting them. Note that clone nodes may not be placed on the clipboard.

If a graph window is active, the selected text is cut to the clipboard. There are separate clipboards for nodes and text. Therefore, cutting or copying an item of one type does not always remove an item of another type from the clipboard.

Copy...

Copies the selected portion of the active window onto the appropriate clipboard. With limited exceptions, it has the same functionality as the Cut command, described above. The principal difference is that the Copy command does not delete the selected material from the active window, but simply places a copy of it on the clipboard.

Copy Special...

Copies the selected portion of the active window onto the clipboard in a format other than the standard TreeAge Pro format. All TreeAge Pro's documents (trees, influence diagrams, and graphs) will allow you to copy the document in bitmap or metafile formats. See Chapter 4. When a tree window is active, you may copy a value calculated at the selected node as a DDE link. When a graph window is active, you may copy the graph data as spreadsheet-accessible text. This functionality is also present in the Graph > Text Report dialog. See Chapter 5. When a tree is rolled back and custom columns are displayed at end nodes, this menu command will copy the columns to the clipboard. The text may then be pasted into a text document or spreadsheet. See Chapter 11.

Paste

The inverse of "Copy," this command copies the contents of the clipboard to the selected point in the active window. For example, if a tree window is active and there is a subtree on the active tree clipboard, the subtree is pasted onto the selected node(s). Tree clipboards are maintained separately from all other types of clipboard contents. When the selection is unambiguous, TreeAge Pro will automatically paste the item of the appropriate type into the active window. When it is unclear whether text or a tree component is to be pasted, a dialog box will appear asking the user to make the selection.

Paste Link

Allows you to paste an item, which is dynamically linked to a spreadsheet, database or other tree, into the active tree. Paste Link is only available when the cursor is in a value field (such as a variable definition or node probability) in a TreeAge Pro tree window. Choosing this menu item eliminates the step of opening the Dynamic Links dialog before pasting the link. See Chapter 20, for more information.

Clear

Deletes the selected portion of the active window altogether. It has the same functionality as "Cut," but the material being cleared is deleted without being transferred to the clipboard, and any material on the clipboard is unaffected.

Links

Manages the sharing of data between trees or from another application to a tree, using either Dynamic or DDE Links. The Links dialog is an alternative to the Paste Link method, described above, of creating dynamic links in a tree.

Break Link…

Destroys the DDE server link(s) to calculated node values at the selected node.

Create Clone…

Designates the selected subtree as a clone master, which may be replicated at different points throughout the tree. Changes subsequently made to the clone master will "ripple through" the tree, affecting all copies of that clone. When a clone master is selected, this command will be entitled "Destroy Clone." See Chapter 21 for more information on clones.

This command is also used to create (and destroy) clones in an influence diagram window. See Chapter 27 for more information.

Attach Clone…

Attaches to the selected node a dynamically-linked copy of a clone master, which is designated by the user from a list of all clone masters previously created in the active tree. When a clone copy is selected, this command will be entitled "Detach Clone." See Chapter 21.

Clones…

Allows you to manage a list of clones in a tree. You may rename or renumber your clones. You may also destroy clones from this dialog, rather than from the Destroy Clone menu item.

Tree Clipboards 1…4

For convenience in working with large trees, there are four distinct "tree clipboards" which can be used in cutting, copying, and pasting subtrees. Tree clipboards are maintained independently of other types of clipboard contents, such as text, annotation boxes, or influence diagram nodes.

The clipboard with the checkmark next to it is designated the active clipboard; when a tree clipboard is not empty, the type of content is designated in parenthesis. For example, when a tree is in the active window and a subtree is selected, choosing "Cut," "Copy," or "Paste" invokes the active clipboard. Thus, cutting a subtree with Tree Clipboard #1 active puts the subtree on Tree Clipboard #1, and pasting with Tree Clipboard #2 active pastes the contents of Tree Clipboard #2 into the tree. By switching the active clipboard (accomplished by selecting the menu item), you can keep up to four commonly appearing subtrees in clipboards from which they can be pasted at will.

Switching the active clipboard does not make any immediate or automatic changes to either your model or the contents of the four clipboards.

Show Tree Clipboard

Displays the contents of the currently active tree clipboard. No editing may be performed in the Tree Clipboard window.

Preferences…



Displays the main Preferences dialog box. See Appendix C for more information.

Numeric Formatting…

Allows you to change the presentation of numeric values in trees and graph windows. See Chapter 5 for more information.

Display menu

Create Note…
Allows you to annotate a tree or an influence diagram. After choosing this command, a rectangular box can be drawn in the window and the annotation typed within it. The box containing the annotation can be moved or resized. Notes can be cut, copied and pasted (see the Edit menu, above). See Chapter 10.

Bind Note…
Allows you to permanently link an annotation box to a particular node in your tree. Notes which are not bound remain floating above the tree, and are not automatically moved when the tree is resized.

Create Arrow…
Allows you to draw an arrow in your tree, as an aid to annotation.

Redraw Window
Redraws the active window to eliminate blank areas and other problems with screen display.

Font
Allows you to choose the font, size, and style in which any selected text will be displayed and printed. Note that you cannot have different fonts, sizes or styles within the same text item. In other words, a node label must be all in one font, size, and style; likewise, an outline must be all in one font, size, and style.

Skip Generation
Forces the selected nodes to display as if they spanned an extra generation. The text will remain at the left edge, while the node's symbol will move to the right so that it lines up with the symbols of the next generation. The primary use of this feature is in asymmetrical trees where it is desirable to have all events within a given stage or time frame displayed in vertical alignment. A node may be made to skip as many generations as desired. See Chapter 11.

Unskip Generation
Negates the effect of a single Skip Generation command when performed on one or more *skipped* nodes.

Align…
Allows you to align two or more nodes in an influence diagram. See Chapter 26.

Collapse Subtree…
Hides the subtree emanating from the selected node. This is useful when working with large trees, when focusing an audience on a particular location in your tree, or when viewing custom columns at end nodes during roll back. See Chapter 11.

Expand Subtree Once
Expands the collapsed subtree at the selected node. Only the first generation will be displayed; the subtrees emanating from the nodes in the first generation will remain collapsed.

Expand Entire Subtree
Reverses the effect of Collapse Subtree.

Suppress Clone Updating
Temporarily turns off the updating of clone copies. See Chapter 21.

Zoom In
Displays your document at a larger magnification.

Zoom Out
Displays your document at a smaller magnification.

Zoom…
Allows you to set the magnification factor for the active window. See Chapter 2.

Show/Hide Explorer View
Shows the explorer view in a tree window. See Chapter 2.

Show/Hide Properties Grid
Shows the properties grid in a tree window. See Chapter 2.

Show Debug Output
See Appendix C, Debugging Preferences.

Show Finder
See Chapter 9, Find/Replace.

Show Node Outline
See Chapter 21.

Show Tool Bar
Shows the tool bar at the top of the screen. Icons on the tool bar allow you to issue various commands with a single mouse click. When the tool bar is visible, this command reads “Hide Tool Bar.” A full tool bar reference is included at the beginning of this appendix.

Show Status Bar
Displays, at the bottom of the screen, a one-line explanation of the selected menu or tool bar item. In addition, the status bar displays some useful information about the active tree. When the status bar is visible, this command reads “Hide Status Bar.” A full status bar reference is included at the beginning of this appendix.

Values menu

Variables and Tables...



Displays a dialog box in which you can create, modify or delete variables and tables. See Chapters 13 and 17 for more information on variables and tables, respectively.

Insert Variable…

Allows you to insert a previously-created variable into the definition of a probability or another variable.

Formula Editor…



Allows you to edit a formula, and insert it into the active text editor if there is one.

Multi-Attribute Weights…

Allows you to set the formula used for generalized multi-attribute calculations. See Chapter 20.

Change Payoff…

Allows you to change the payoff (variable, expression, or value) associated with the selected terminal node(s).

Markov Termination… (Healthcare module only)



Allows you to specify the condition(s) upon which a given Markov process will cease to cycle.

Markov State Information… (Healthcare module only)



Accepts values for the initial, incremental, and final rewards associated with a given state of a Markov process. You may also indicate that the selected state should be an automatic tunnel state.

Markov Transition Rewards… (Healthcare module only)



Assigns a reward to a non-state node in a Markov process.

Open Calculator/Evaluator



Displays the Calculator/Evaluator dialog box, which allows you to enter an expression whose value you wish to calculate at the selected node. Calculations are performed based on the values of the variables in effect at the selected node. See Chapter 14.

Create Slider…

Allows you to edit manually the value of a variable at a given node. After choosing a variable and specifying a range, drag the sliding "thumb" to change temporarily the value of the variable. The changed value will remain in effect until the slider is closed, or until a sensitivity analysis is run on the chosen variable. See Chapter 14.

Show Variable Definitions Window…



Displays the list of variables which have been defined at a selected node. Node selection can be changed by using the arrow buttons in the window. Double-clicking on a variable in the window brings up a Define Variable window for editing the definition at the selected node. When the Variable Definitions window is displayed (whether or not it is the active window), the menu command will read "Hide Variables Window." See Chapter 14.

Distributions…

Allows you to assign parameters to an analytic distribution for use in a payoff formula or probability. Distributions are used primarily for sampling during Monte Carlo simulation. See Chapter 16.

Probability Wheel…



Displays the probability wheel at the selected chance node. The wheel is used to assign subjective probabilities. See Chapter 12.

Options menu

Select Subtree

If the currently selected node has branches, this option causes the entire subtree rooted at this node to be selected. This is useful in cutting or copying a subtree, which requires selection of the entire subtree. It can also be used to select the tree (i.e., the subtree emanating from the root node), prior to copying the tree to the clipboard, either in TreeAge Pro format or for export in bitmap/metafile format.

Select If…

Allows you to select nodes using a rule. For instance, you may select all terminal nodes, or all nodes at which a particular variable is defined. See Chapter 8.

Find…

Searches for (and replaces, if desired) specified text in the tree window. See Chapter 12.

Node Comment…



Allows you to assign an annotation to the branches of the selected node. This feature is often used to record the reasons underlying probability assignments. See Chapter 10.

Add Branch(es)

Adds two branches to the currently selected node. (You may change the default number of branches in the Preferences dialog.) If the selected node already has branches, this command will be entitled "Add Branch" and will add one additional branch to the selected node. A chance node is automatically added at the end of each new branch; you may later change the node type.

Insert Branch…

Inserts a new node to the left, to the right, above, or below the selected node. See Chapter 5 for more information.

Delete Branch…

Deletes the selected branch from the tree, and attaches any branches emanating from the deleted branch to its parent.

Reorder Branches…
Allows you to change the top-to-bottom ordering of the branches emanating from a node. See Chapter 5.

Show/Hide Markov Bindings…
At the selected node *within a Markov subtree* (including the Markov node), allows you to associate a state name variable with a Markov state. These *Markov bindings* can be used in place of a Markov state name when assigning the transition for a Markov transition node (see below). Markov state bindings are useful when cloning within Markov subtrees. Refer to the Healthcare module documentation for details.

Markov Jump Node… (Healthcare module only)
For the selected Markov transition node, allows you to choose from a list of existing Markov states (within the same Markov process) and Markov state bindings. Refer to the Healthcare module documentation for details.

Change Node Type…

Presents a dialog box which allows you to change the type of the selected node(s). In the influence diagram window, the Change Node Type command is found under the Diagram menu; see below.

Change Optimal Path…
When one or more decision nodes are selected, allows you to change the optimal path for the selected decision node(s) to be the opposite of the default for the rest of the tree. See Chapter 12 for more information.

Force Path
This option allows you to indicate the occurrence (or inevitability) of a particular event at a chance node, or of a commitment to a specific alternative at a decision node. TreeAge Pro will change the selected node to a logic node and set the logical expression of the specific event or alternative to 1 and the logical expressions of the other branch(es) to 0.

Enter Risk Preferences…
Accepts information about the risk aversion of the decision maker for application of a risk preference curve to the decision at hand. See Chapter 24.

Distribute Children…
Approximates a continuous distribution of values by creating a specified number of branches and assigning a particular value to a variable at each of those branches, according to a distribution which you chose and parameterize. See Chapter 22.

Bayes' Revision…
Uses Bayes' theorem to revise probabilities in a subtree to reflect knowledge gained from an imperfect predictor of an uncertainty. The *a priori* and likelihood probabilities are automatically converted to the probabilities needed to make a decision based on the outcome of the imperfect test. All values can be preserved as variables for purposes of sensitivity and threshold analysis. See Chapter 23.

Show Custom Interface…
Displays the window associated with the Basic Custom Interface, as described in Chapter 25. When the Custom Interface is shown, this command will read "Hide Custom Interface."

Design Custom Interface…
Allows you to design a basic or extended Custom Interface. The Custom Interface is useful for sharing models with less sophisticated users. See Chapter 25.

Mimic Run-Time
Causes TreeAge Pro to emulate the Run-time TreeAge Pro software in certain respects. This will affect menu items and preference settings, many of which are not available to run-time users. Use this command to test an Extended Custom Interface tree.

Analysis menu

Roll Back



This option does all the basic calculations on the active tree and causes the results to be displayed in the tree window. See Chapters 1, 2, and 5 for more information on roll back and changing what the tree calculates, respectively. Choosing the Roll Back option again returns the tree to its normal display, and the information described above disappears.

Sensitivity Analysis





Tests the sensitivity of a recommended decision to changes in the value of one or more variables across a range (or ranges) specified by you. One-way sensitivity analysis and tornado diagrams are also available when a chance node is selected. Depending on the circumstances, calculations may be done on the basis of expected value and marginal value. See Chapters 13 and 15 for more information.

Monte Carlo Simulation



Performs a Monte Carlo simulation at the subtree rooted at the selected node. See Chapter 16 for information on different types of simulation.

Probability Distribution / Comparative Distributions



Draws a bar graph that displays the distribution of payoffs in terms of their probability of occurrence. If a decision node (or multiple nodes) is selected, this option reads Comparative Distributions. See Chapter 5 for more information on these analyses.

Rankings…

Displays expected value of the alternatives associated with a decision node, ranked in order of optimality. See Chapter 5.

Markov Cohort Analysis… (Healthcare module only)

Allows you to graph, for a given Markov process, many of the quantities which vary over the course of the process. You may also view a full “trace” of the process, with all values displayed in a table. Refer the Healthcare module documenation for details.

Cost-Effectiveness… (Healthcare module only)

For a selected decision node in a cost-effectiveness tree, displays each alternative in a graph with increasing cost to the right and increasing effectiveness toward the top of the graph. This analysis will also show dominance and extended dominance, and all marginal values are available via the Graph > Text Report command. Refer the Healthcare module documenation for details.

Net Benefits… (Healthcare module only)

For a selected node in a cost-effectiveness tree, displays each alternative's changing Net Benefit as willingness-to-pay increases from left to right. This analysis will show changes in optimal strategy as the weight on effectiveness increases. Refer the Healthcare module documenation for details.

Graph Risk Preference Function

Graphs the currently active risk preference function as a line graph; see Chapter 24.

Threshold Analysis…

Searches more thoroughly and accurately for threshold information in connection with a single variable than does a one-way sensitivity analysis. The result of this analysis is a detailed, textual description of how the optimal strategy is affected by changing the value of a single variable across a designated range. See Chapter 15.

Over/Under…

Allows you to specify a target value at a selected node, and then calculates the probability of an outcome having a value over the target, and the probability of an outcome at or under the target. See Chapter 5.

Expected Value

Displays the expected value of the subtree rooted at the selected node; see Chapters 1 and 2. If multiple nodes are selected, TreeAge Pro will calculate and display the sum of the expected values at these nodes.

Expected Value of Perfect Information

Calculates the maximum value of eliminating an uncertain outcome. See Chapter 5.

Path Probability

Displays the probability of reaching the selected node, that is, the probability that the scenario represented by the path between the selected node and the root node will occur. If multiple nodes are selected, TreeAge Pro will calculate and display the sum of the path probabilities.

Payoff Range

Determines the unweighted highest and lowest potential payoffs in the subtree rooted at the selected node. See Chapter 5.

Standard Deviation

Calculates the standard deviation of the potential outcomes at a selected chance node. The standard deviation is weighted by the probabilities of the branches. It indicates the extent of dispersal, around the expected value at the selected node, of the values in the branches departing from that node. This gives an indication of the risk involved in the subtree rooted at the selected chance node. For information on exporting the standard deviation, see Chapter 5.

Show Optimal Path

Specifies the branch emanating from the selected decision node which represents the best choice that can be made. This information is also displayed when you select Analysis > Expected Value at a decision node.

Verify Probabilities

This command is used to check every chance node in the active tree to determine whether all sets of probabilities sum to 1.0. An appropriate message specifying the location of any error, or indicating a successful verification, appears at the end of the verification process.

Storage…

Enables you to store the parameters of an analysis that you just ran, recall and rerun a stored analysis, or delete or rename a stored analysis. See Chapter 19.

Diagram menu (influence diagram window)

Description…

Allows you to annotate a node in an influence diagram.

Variable…

Makes it possible to rename the variable which holds the value for a node in an influence diagram. The variable may then be used in a payoff formula at a value node. See Chapter 26.

Outcomes / Alternatives…

This command is used to enter a set of outcomes (for a chance node) or alternatives (for a decision node) in an influence diagram. Its name will change depending on the type of node selected. See Chapter 3.

Probabilities…

Enables you to assign the conditional probability distribution(s) associated with the outcomes of a chance node in an influence diagram. See Chapter 3.

Values…

Makes it possible to assign the conditional value distribution(s) to a chance or decision node in an influence diagram for use in a payoff formula. This menu item can also be used to enumerate payoff values for a value node. See Chapter 3.

Straighten Arc

If an arc in an influence diagram has been curved (by dragging the black selection handle), this command can be used to straighten it. See Chapter 26.

Flip Arc

Changes the direction of the selected arc in an influence diagram. This operation is particularly useful for performing EVPI calculations, as described in Chapter 26.



Change Node Type…

Enables you to change the type of a node in an influence diagram. There are four types of representative nodes: decision nodes (squares), which

indicate a decision to be made; chance nodes (circles), which indicate an event over which the decision maker lacks complete control; value nodes (diamonds), which must be used for the model's final outcome and may be used to create an intermediate formula; and deterministic nodes (circle with double outline), which are used to specify a parameter having a fixed value and, optionally, a value range for purposes of sensitivity analysis.

Bayes' Revision…

Identifies, in a text report, any nodes at which Bayes' revision will be performed when the influence diagram is converted into a tree. See Chapter 27.

Graph Actions menu

Text Report…

Displays the numerical data which underlie the graph in the active window. These data may then be exported for further analysis or graphing in a spreadsheet, statistics, or database program. See Chapter 6.

New Label…

Adds a new, custom label to the active graph.

Show/Hide…

Allows you to hide certain pieces of information from the selected graph. You may hide lines in a line graph or bars in a tornado diagram.

Distribution Statistics…

Displays the basic statistical values associated with a distribution graph.

Line to Table…

Converts a line in an active line graph into a table.

Strategy Graph

Graphs the optimal frontier of a one-way sensitivity analysis graph.

Create Template…

Stores the layout of the active graph window for later use in other graphs. See Chapter 6.

Apply Template…

Applies an existing graph template to the active graph window.

Maintain Templates…

Allows you to edit the information stored in specific graph templates.

Options…

Opens a dialog which allows you to change the contents and appearance of the graph. Many options are specific to certain graph types and are explained in the appropriate analysis chapter or in Chapter 6.

Table Actions menu

Add Entry…

Adds a new entry to the open table. See Chapter 17.

Edit Entry…

Edits the index/value associated with the selected entry in the open table. See Chapter 17.

Delete Entry

Deletes the selected entry in the open table. See Chapter 17.

Edit in Excel [Excel module only]

Edits the properties (name, file name, lookup method) for the open table in a new Excel worksheet. See Chapter 29.

Load Text File…/Save Text File…

Import or export table values. See Chapter 17.

Copy Tables/Paste Table…/Clear Table…

Paste and Clear prompt you first to clear the existing table data. See Chapter 17.

Graph Table

Displays the contents of the open table as a line graph.

Properties

Edits the properties (name, file name, lookup method) for the open table. See Chapter 17.

Appendix C

Preferences

This appendix describes TreeAge Pro's Preferences dialog.

In this chapter:

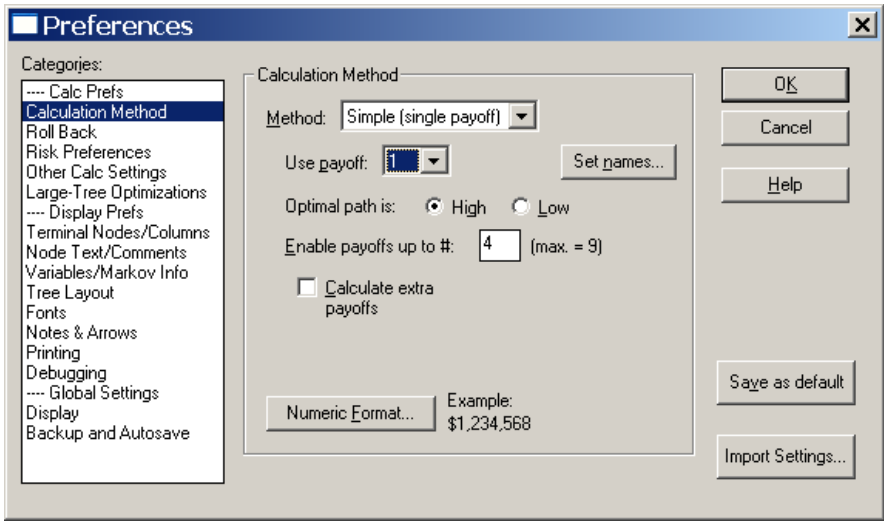
- Using the Preferences dialog..... 558
- Tree preferences 559
- Influence diagram preferences 571
- Global preferences 573
- Backup and Autosave settings..... 574

Using the Preferences dialog

The Preferences dialog, which controls many settings and options in the tree and influence diagram windows, can be found under the Edit menu (or by pressing F11). The list of available categories changes depending on what, if any, type of document is currently active. Choosing a category from the list changes the page of options that appear in the center frame. The options in each category are described in this appendix.

The Global Settings at the bottom of the list are, as the name suggests, program-wide preferences (not document-specific). Changes to the other preferences apply to the *currently-active document*.

You have the option of saving a document's current preferences as the default settings, to be automatically applied to all new documents of that type. Simply click the **Save as default** button before clicking OK to close the dialog box. (Note that all available categories for the document are saved as default, not merely the currently selected category.)



In a tree window, you also have the option of importing preferences from an existing, saved document (overwriting the tree's current preferences). Click the **Import Settings…** button and locate the source document using the Open File dialog.

Hot keys when opening the Preferences dialog

The CONTROL and SHIFT keys also have special functions when the Preferences dialog is being opened, allowing preferences to be reset.

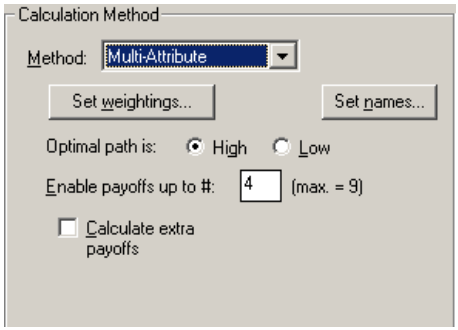
CONTROL = reset the current tree's preferences back to the program defaults

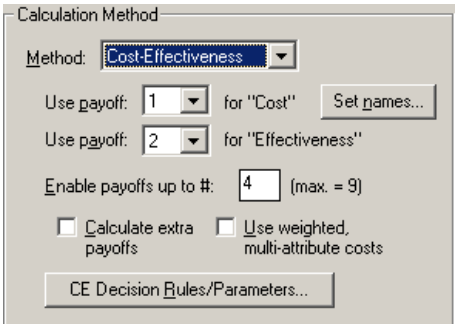
SHIFT = reset tree and default preferences back to program defaults

Calculation method preferences (tree window)

See screenshot of this category on the previous page.

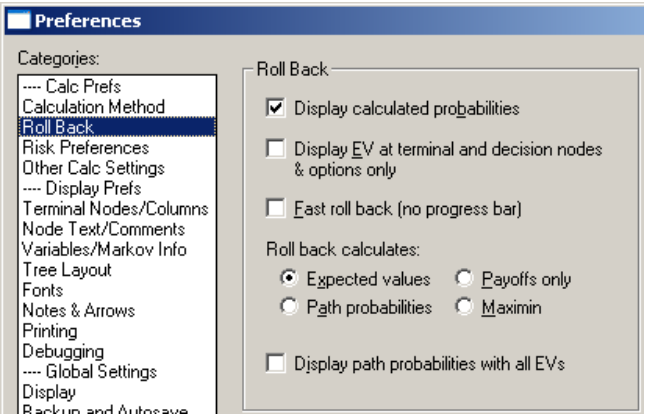
Option	Description
Method	The drop-down list box specifies the calculation methods available in TreeAge Pro: Simple, Benefit-Cost, and Multi-Attribute; changing the selection changes some of the other calculation options. See Chapter 7 for more calculation methods.





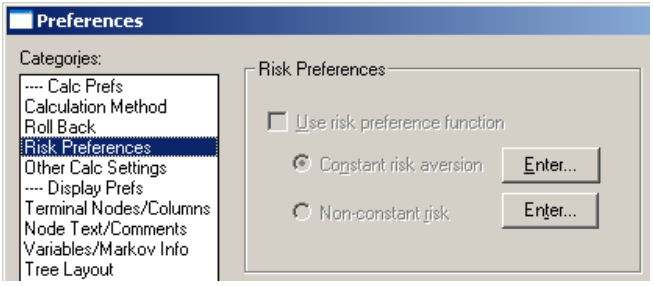
Set names…	Custom labels can be assigned to the nine payoff sets in a tree. The custom labels/names are used to label payoff and reward editors in the Enter Payoff, Properties Grid, Markov Rewards, and Multi-Attribute Weights windows.
Use payoff	You can assign up to nine payoffs at each terminal node. Changing the "Use payoff" drop-down selection determines which of these payoffs are to be used in calculating the active tree.
Set weightings…	If the Multi-Attribute calculation method is active, you must set weightings for each of the payoff sets you use. The Use payoff drop-down list box will be replaced by the Set weightings… button.
Optimal path is	When Simple or Multi-Attribute calculations are active, select the appropriate optimal path. Select High when your tree is to be calculated on the basis of profit, utility, cash flow, quality of life, or other criterion that should be maximized. Select Low when payoffs are costs or other attributes which should be minimized. Separate optimal path settings are maintained for each multi-attribute calculation method and, for the Simple calculation method, each of nine possible payoff sets.
Enable payoffs…	TreeAge Pro allows you to activate fewer than all nine payoffs, in order to simplify/clarify the Enter Payoff dialog and the multi-attribute Weightings dialog.
Calclulate extra...	TreeAge Pro can calculate non-active payoffs (e.g., payoffs 2-4 during Simple calculations using only Payoff1).
Numeric format	The Numeric Format… button is used to set the format used to display calculated values. As with the Optimal Path setting, there are individual numeric formatting options for each Simple payoff, and each multi-attribute method. See Chapter 5 for more information on numeric formatting.

Roll back preferences (tree window)



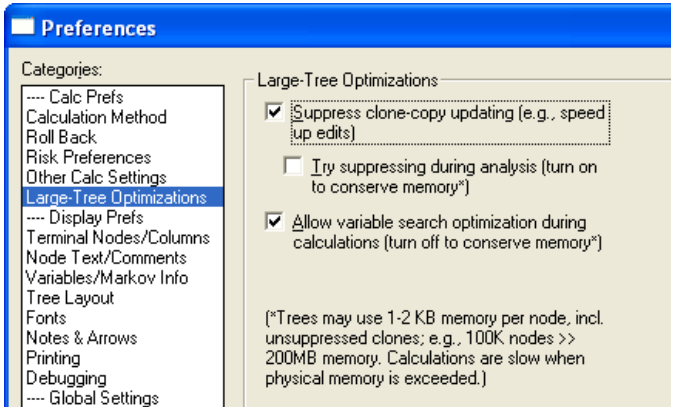
Option	Description
Display probabilities as numeric equivs	This option relates primarily to probabilities which have been entered as variables. If the option is turned on (the default setting), the calcualted value of each of the probabilities will be displayed using numeric formatting preferences during roll back. Turning off roll back will show the original, uncalculated probability expression.
Display EV at terminal and decision nodes & options only	If this option is turned on, roll back boxes will displayed only at terminal nodes, decision nodes, and branches of a decision node.
Fast roll back	Normally, a progress indicator will display in the status bar while a tree is rolling back. Selection of the Fast roll back option will suppress this display, resulting in speed increases of up to 100%, depending on the size and complexity of the tree. With either option, you may cancel calculations by pressing ESC. The four roll back calculation options are described in Chapter 11.

Risk preferences (tree window)



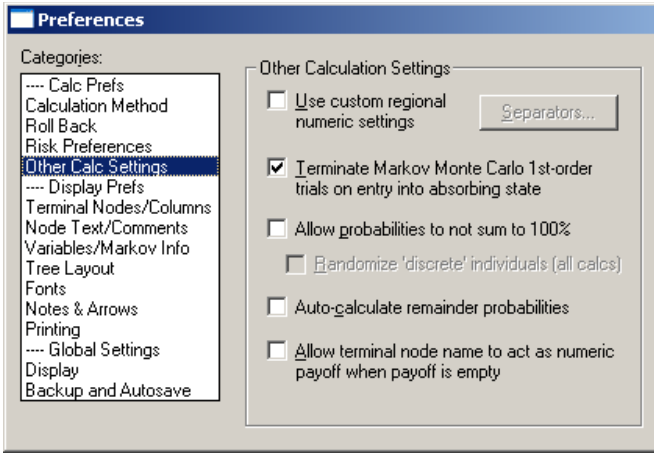
Option	Description
Use risk preference function	When this box is checked, TreeAge Pro's calculations will be based on a risk function rather than expected value. If this option is dimmed, a risk preference function has not previously been entered for the active tree. See Chapter 24.
Constant risk aversion	When this option is selected, calculations will be based on a constant risk aversion function, rather than expected values.
Non-constant risk	When this option is selected, calculations will be based on a non-constant risk aversion function, rather than expected values.
Enter…	There are two Enter buttons, one for a constant risk aversion function and the other for entering a non-constant risk aversion function. These two functions and their differences are described in Chapter 24.

Large-Tree Optimizations preferences (tree window)



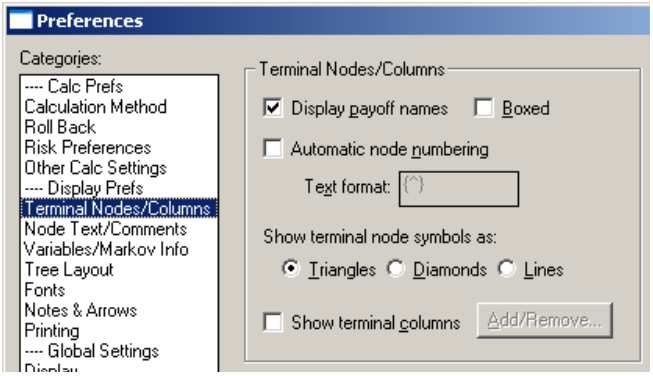
Option	Description
Suppress clone-copy...	See Chapter 21 for details on clones and clone suppression. Note that this setting is duplicated in the Edit > Clones… dialog.
Try suppressing during analysis... **Allow variable search optimization...**	These two settings can be used to greatly limit the amount of memory required for analysis of heavily cloned trees. In trees that are not currently using up available physical memory, however, these settings will slow down analysis somewhat.

Other calculation preferences (tree window)



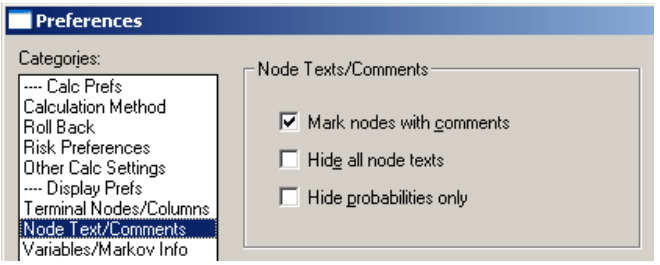
Option	Description
Custom regional numeric settings	Use this option when you want the calculation preferences specified in your tree to override the regional numeric settings in Windows. Selecting this option and clicking the Set button opens the Custom International Settings dialog, where you can specify which characters a particular tree should use for decimal and thousands separators. See Chapter 5.
Terminate Markov Monte Carlo simulations on entry into absorbing state	By setting this option, you indicate that the termination conditions should be ignored during Monte Carlo simulations of a Markov process. See the Healthcare module documentation for more information.
Allow probabilities to not sum to 1.0	When this option is checked, calculations will not stop with errors when a chance node's branches do not sum to 1.0. See Chapter 22 for important information on this settings, and its sub-options.
Calculate complementary probabilities automatically	When this option is checked, TreeAge Pro will fill in the last probability in a set of branches emanating from a chance node, so long as all the other probabilities on branches emanating from that node are wholly numeric (i.e., no variables are used).
Allow terminal node name to act as numeric payoff	It is possible to have TreeAge Pro treat the branch description at a terminal node as that node's numeric payoff value.

Terminal nodes/columns display preferences (tree window)



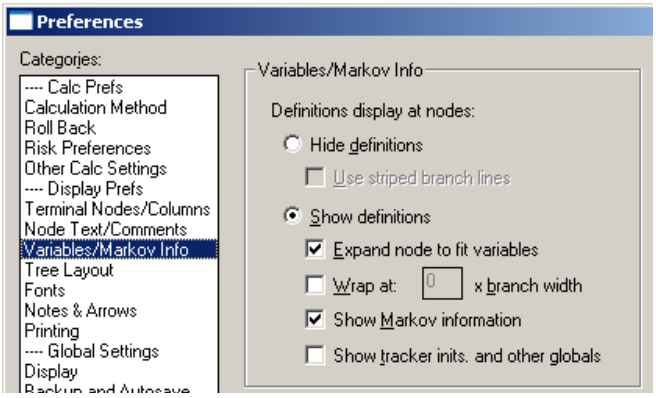
Option	Description
Always display payoff names	When this option is on, each terminal node will display the name of its then active payoff. This can be helpful for identifying terminal nodes where a payoff has not been assigned. In the case of trees having multiple payoffs, this feature makes it possible to see at a glance which payoff is active.
Boxed	If you have chosen to always display payoff names, this option lets you choose whether the payoff names should be enclosed in a box. This option relates only to tree display prior to roll back; during roll back, calculated values are always boxed.
Automatic node numbering	If this option is on, terminal nodes will display the custom text entered in the field. Use the ^ (caret) symbol in the text to represent the scenario number.
Show terminal nodes as	TreeAge Pro can show terminal nodes using any of the three methods shown. Triangles are the default (and standard) method. Diamonds are used to indicate the parallelism between terminal nodes in a tree and value nodes in an influence diagram. Lines are for those applications when you do not want any symbol displayed to the right of a final outcome.
Show columns	Selecting this option will allow you to display custom columns of values to the right of the rolled-back tree. Clicking the Set… button opens the Terminal Node Columns dialog, where you can choose the calculated values (including expected and marginal values, as well as custom calculations) and formats you want. See Chapter 11.

Node text/comment preferences (tree window)



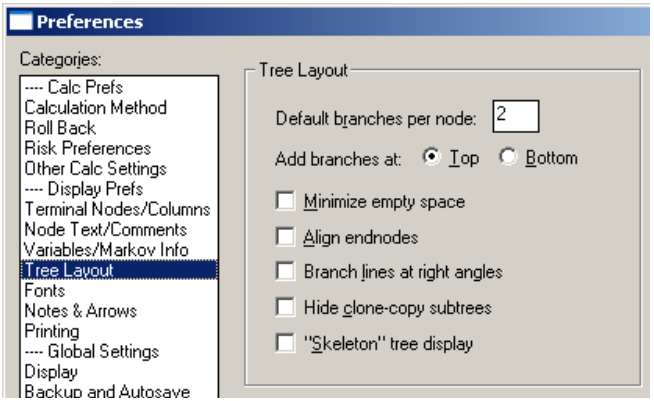
Option	Description
Mark nodes with comments	If this option is on, nodes at which you have entered a Node Comment (see Chapter 10) will be displayed with a small flag above the symbol. This flag does not print or export.
Hide node texts	When this option is on, the display of all textual information in the tree window is suppressed. Use this flag to get a picture of the structure of your tree.
Hide probabilities only	When the Hide node texts option is *off*, this option is available. The display of probabilities is suppressed in the tree window, while all other textual information is visible. Use this flag to temporarily simplify the display of complex trees with many uncertainties.

Variables display preferences (tree window)



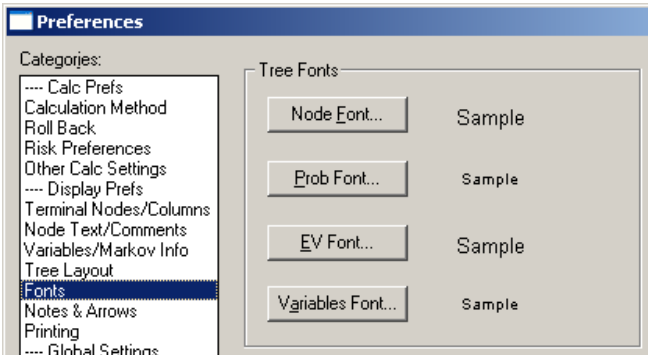
Option	Description
Hide definitions, Use striped branch…	This setting will not show nodes' variable definitions (with optional marking of branch lines).
Show definitions (Expand node to fit…, Wrap at: N x branch…, Show Markov…)	When Show definitions is selected, long definitions will be clipped to the natural length of the node. Options are: to force node lengths to expand to fit the definitions or wrap definitions at a fixed width; and to show Markov quantities — termination conditions, rewards, and Markov bindings — in the tree.

Tree layout preferences



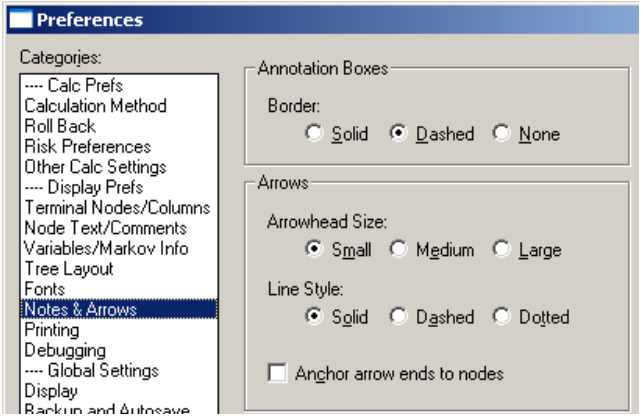
Option	Description
Default branches per node	This option sets the number of branches which are added to a node when you select Add Branches from the Options menu. The default number applies only the first time that branches are added at a given node. Once a node has branches, additional branches are added one at a time.
Add branches at…	This option enables you to control whether additional branches are to be added above or below existing branches. The Insert Branch command under the Options menu provides additional flexibility in this area. See Chapter 9 for more information.
Minimize empty space	Use of this option produces a "compressed" version of your tree. No vertical space is wasted. Because each node no longer has its own horizontal "slice" of the tree display, this option may not be used with Align endnodes. See Chapter 11.
Align endnodes	Forces all terminal nodes to line up at the rightmost edge of the tree.
Branch lines at right angles	Normally, branch lines are drawn at whatever angle is needed to provide the most direct connection from one node to the next. When this option is on, all branch lines are drawn vertically, then horizontally, rather than obliquely.
Hide clone-copy subtrees	Suppresses the display of clone-copy subtrees; display of clone masters is not affected. When this option is selected, only the name of the clone is displayed to the right of the copy node. Use of this option can dramatically reduce the physical size of your tree. See Chapter 21.
"Skeleton" tree…	This setting allows an abstract view of the tree to be displayed or printed, with space inserted between generations, and collapsed subtrees being completely hidden. Used in combination with the Minimize empty space and Hide clone-copy settings, this creates a highly compact view of the tree. See Chapter 11.

Font preferences (tree window)



Option	Description
Node Font	This option enables you to select the default font used for naming nodes (branch descriptions). The font selected in this manner will apply to any new nodes created in the active tree and to any existing nodes, except for any nodes where the font has been set individually.
Prob Font	This option enables you to select the font used in the probability fields of the active tree, in both the rolled-back and unrolled-back state. This allows you to clearly distinguish between probability variable names and adjacent node descriptions.
EV Font	This option enables you to select the font used in the expected value boxes generated during roll back. It also applies to other information which is not user-editable and is displayed next to a node, such as clone names when clones are hidden, or payoff names when Always show payoff names is selected.
Variables Font	This option enables you to select the font used to display the definition of variables when that option is selected in the Variables Display page of the Preferences dialog.

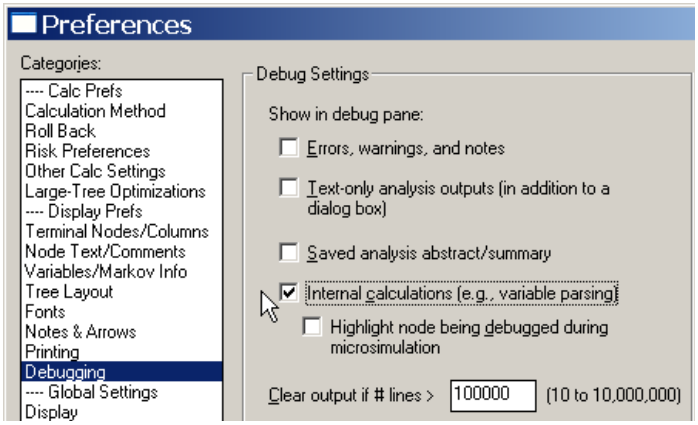
Notes & arrows preferences (tree window)



Option	Description
Annotation Box Borders	This option enables you to specify the type of border which surrounds annotation boxes in the active tree. The options are a border drawn with either a solid line or a dashed line or, alternatively, no border. Your selection will apply to every annotation box in the active tree, bound or unbound.
Arrowhead Size	When your tree contains one or more arrows, this option enables you to choose whether they should be drawn with large, medium or small arrowheads. Your selection will apply to every arrowhead in the active tree.
Arrow Line Style	When your tree contains one or more arrows, this option enables you to choose whether they should be drawn with solid, dashed or dotted lines. Your selection will apply to every arrow in the active tree.
Anchor arrow ends...	This setting will cause arrows with an end close to a node to move and grow as the tree grows.

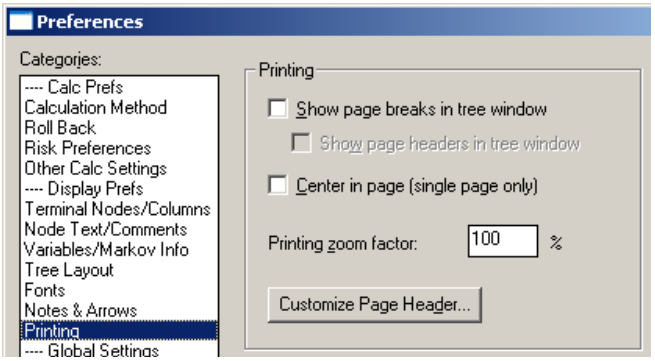
Debugging preferences (tree window)

The Debug pane can be used to display/report a variety of textual information, including error messages, text-only analysis output, stored analysis summaries, and detailed internal calculation (during analysis).



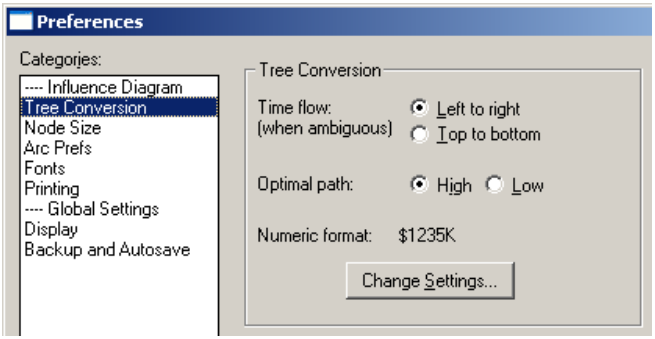
Option	Description
Errors, warnings…	Use this option to avoid having TreeAge Pro pop-up dialogs when warnings or error messages need to be displayed, instead having the text added to the Debug pane.
Text-only analysis	If this option is selected, TreeAge Pro will copy to the Debug pane the outputs of analyses that only report textual results (no graphs), such as Expected Value and Rankings analyses.
Saved analysis abstract	When TreeAge Pro completes an analysis, in most cases the parameters provided to run the analysis can be stored for later reuse (see Chapter 19 on the Analysis > Storage... submenu). If this option is selected, the Debug pane will display summary information about any new analysis available for storage.
Internal calculations…	This is the most commonly useful Debug pane option. If it is selected, TreeAge Pro will output very detailed, step-by-step results of all tree calculations. This includes variable and function evaluation, payoff and probability calculations, distribution and table references, and more. A sub-option under Internal calculations, Highlight node being debugged…, is designed to allow the user to pause calculations at intervals during a microsimulation. For example, at each step in a random walk, each variable calculation, etc., the debugging output could be verified in the debugging pane, in order to check for calculation correctness in complex models. See Chapter 36 for details on using microsimulation.

Printing preferences (tree window)



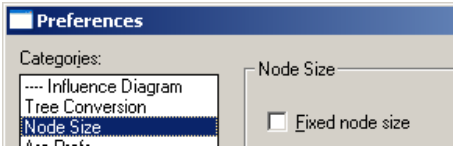
Option	Description
Show page breaks in tree window	If this option is on, the screen display will include dotted lines to indicate where a new printed page will begin. This is likely to be more accurate than page break information shown in the print preview window.
Show page headers in tree window	If this option is on, the screen display of the tree will include any page header or footer that will be included in a printout of the tree.
Center in page	This is one of two ways in which it is possible to determine the location of documents in printouts. This option will apply only to documents which are sized to fit on a single page. When selected, the printout of the one-page document will be centered on the page. If more than one page is required for the printout, selecting this option will have no effect. The alternative method of positioning the tree or graph in a printout is described in Chapter 9.
Printing zoom factor	This option enables you to store a percent reduction/enlargement factor with each document. This scaling factor is document-specific. Note also that it is independent of any scaling specified under Page Setup. Thus, if your printer driver allows scaling via the Page Setup command, you run the risk of applying one percentage against another. The printing zoom factor is also independent of the screen-display zoom factor, set in the Display > Zoom commands.
Page Header	After clicking this button, you will be able to set the page header and footer information for the active tree.

Tree conversion preferences (influence diagram)



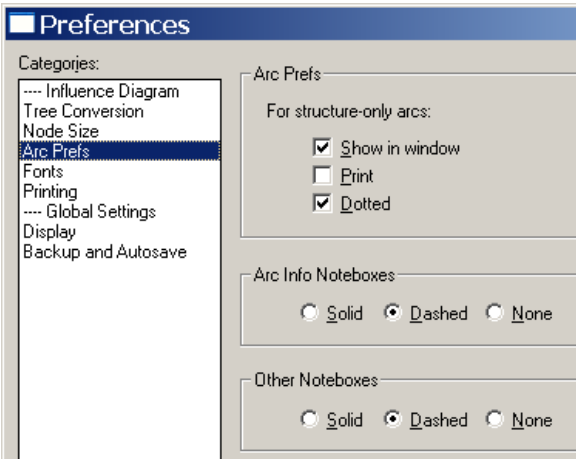
Option	Description
Time flow	The complete algorithm used for converting influence diagrams into trees is discussed in Chapter 25. If the ordering of two nodes is ambiguous, their relative locations in the window are used as a final determination. Select Left to right to have nodes on the left converted before nodes on the right; or select Top to bottom to have nodes toward the top converted before nodes toward the bottom. These options apply only after all other rules have failed to determine the proper node ordering.
Optimal path	While no calculations actually occur in the influence diagram window, you may set this flag to avoid having to reset it in the tree each time you convert the influence diagram. See Calculation Method preferences, above, for details.
Numeric format	This is also carried over to the converted tree. You may enter your numeric formatting preferences to avoid reentering it each time you convert to a tree.

Node size preferences (influence diagram)



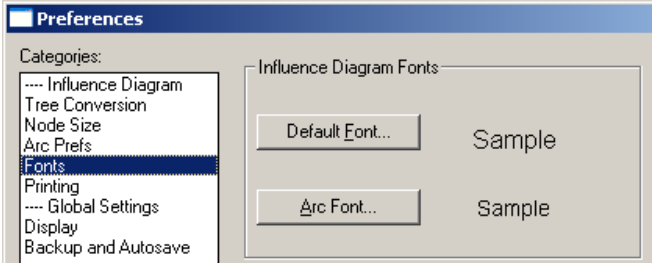
Option	Description
Fixed node size	Some influence diagram users prefer to have each node sized identically, rather than having TreeAge Pro determine the node's size individually, based on the amount of text entered to name the node. If you opt for a fixed node size, you may indicate the size of the text area in the lower area of the dialog. Drag the corner handle to the appropriate size allocated to the text of a node; TreeAge Pro will add extra space for the node's border.

Arc preferences (influence diagram)



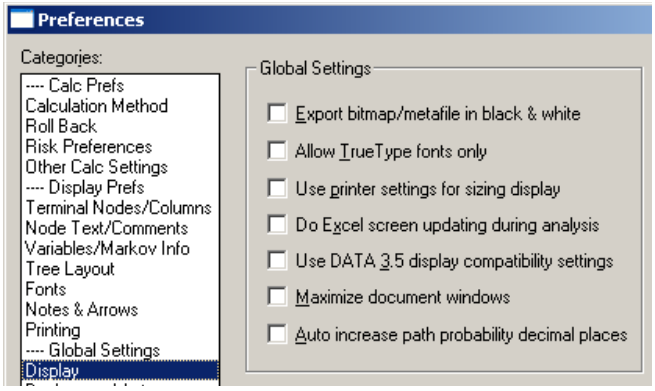
Option	Description
Show in window	Deselect this option to suppress screen display of structure-only arcs altogether. You will not be able to select these arcs for editing until you reselect this option. For information on using structure-only arcs, see Chapter 25.
Print	If this option is selected, structure-only arcs will print with the rest of the influence diagram.
Dotted	Select this option to force structure-only arcs to display or print as dotted lines, rather than as straight lines.
Arc Info Noteboxes	This option allows you to control the line style for notes added in the Arc Info dialog.
Other Noteboxes	This option allows you to control the line style for free notes added in influence diagram.

Influence diagram font preferences



Option	Description
Default font	If you change the default font, this will affect all new nodes in the influence diagram, as well as all other nodes at which you did not set the font individually.
Arc font	This font is used for the comments of all arcs in the influence diagram. You may not set the font for a single arc comment individually.

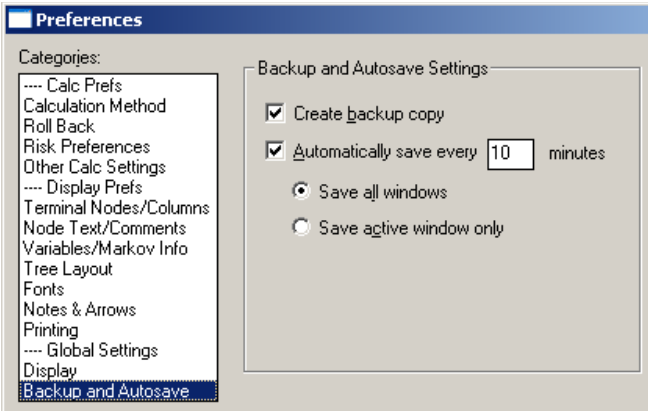
Display settings (global)



Option	Description
Export bitmap/metafile in black & white	Use this option if your exported documents will eventually be printed on a black and white printer.
Allow TrueType fonts only	If this option is on, font-selection dialogs will allow you to select only TrueType fonts, which print and display identically. This selection will *not* affect fonts you have already selected.
Use printer for sizing	Whenever the tree must be resized (such as after you add nodes or change a branch description), TreeAge Pro must calculate the width and height of each body of text. However, the screen and printer do not always agree on the exact amount of space needed. Selecting this option may improve the quality of either the printed output, or the screen display, eliminating problems such as a branch description overlapping the node symbol.

Backup and Autosave settings (global)

TreeAge Pro has automatic backup and autosave capabilities. Backup and autosave preferences are global, not file-specific, and can be customized by choosing **Edit >** Preferences.



Create backup copy

If this preference is selected, TreeAge Pro will save a backup copy of each file that you open and subsequently save changes to. The backup file contains the version of the file as it existed it was opened. No matter how many times you modify or save the file after opening it, its backup file continues to contain the original version of the document. If you close a file and then reopen it, the backup process is restarted; if a backup file already exists for a document that you open, TreeAge Pro will overwrite the existing backup file when you save the document.

The backup file is created in the same directory as the original file, with "-backup" appended to the extension; for example, the backup file for "my tree #1.tre" would be saved as "my tree #1.tre-backup".

If you accidentally save changes to a file that cannot be undone, you can open the backup copy of the file. To do so, however, you will need to change the Open File dialog's "Files of type:" drop-down menu to "All files (*.*)".

In the case of a new, untitled document, a backup file is not created the first time it is saved, since no original document file existed on disk at that point; the backup file will be created the next time you save changes to the file.

Automatically save…

If the autosave option is selected, TreeAge Pro will save copies of open documents at regular intervals without prompting. Only documents that have unsaved changes will be autosaved. As with a backup file, the autosave file is created in the same directory as the original file; in this case, "-autosave" is appended to the file extension.

When a document is closed normally, autosave files are deleted. If TreeAge Pro or your computer closes abnormally, however, autosave files will not be deleted. And if you subsequently open a tree and TreeAge Pro detects a more recent, autosaved version of the tree, you will be prompted to recover the autosaved version.

Appendix D

Functions and Operators

TreeAge uses a standard set of operators, described here. Functions are described in detail in Chapters 14, 21, and 35.

Functions and operators that take more than one argument are sensitive to user errors; see Chapter 13 for information on using the Functions Helper to set up functions with the correct arguments, and using the Variables Report and the Calculator/Evaluator tool to test the result of calculating a function.

In this appendix:

- Operators 576
- Using mathematical, statistical, and other functions 578

Operators

Mathematical expressions used in TreeAge Pro necessarily contain operators (such as addition or multiplication signs) and may also contain functions (such as logarithm or net present value). Described below are the operators and functions available in creating expressions.

Arithmetic operators

These operators perform arithmetic on the values that surround them. TreeAge Pro uses the traditional syntax for expressions, known as *infix* notation. For example, an expression that adds three and seven would be written 3 + 7, rather than 3, 7 +.

Symbol	Example	Explanation
+	x+y	**Addition.** Returns the sum of x and y.
-	x-y	**Subtraction.** Returns the difference between x and y. (Also used for negation, i.e., to denote negative numbers.)
*	x*y	**Multiplication.** Returns the product of x and y.
/	x/y	**Division.** Returns the quotient of x and y.
^	x^y	**Exponentiation.** Returns x to the yth power.
( )	x*(y+z)	**Grouping.** Returns the product of x and the sum of y and z.

Relational operators

These operators return a *true* or *false* value, depending on the veracity of the expression in which they appear. A true value is represented by a numeric 1, a false value receives a numeric value of 0.

Relational operators are useful in many settings: in If() and Choose() functions, in expressions evaluated at logic nodes, and in a Markov termination condition.

Symbol	Example	Explanation
<	x < y	**Less than.** Returns true if x is less than y, and false if x is greater than or equal to y.
<=	x <= y	**Less than or equal to.** Returns true if x is less than or equal to y, and false if x is greater than y.
>	x > y	**Greater than.** Returns true if x is greater than y, and false if x is less than or equal to y.
>=	x >= y	**Greater than or equal to.** Returns true if x is greater than or equal to y, and false if x is less than y.
=	x = y	**Equals.** Returns true if x equals y, and false if x is not equal to y.
<>	x <> y	**Not equal to.** Returns true if x is not equal to y, and false if x equals y.

It is also possible, using the appropriate relational expression syntax, to test one value in terms of two others. There are a number of acceptable forms, with the two basic ones being:

y < x < z	Returns true if x is both (a) greater than y and (b) less than z.
y > x > z	Returns true if x is both (a) less than y and (b) greater than z.

Other valid forms of this syntax can be created by substituting ">=" for ">" or "<=" for "<" (for example, expressions of the form "y <= x < z" and "y >= x >= z" are valid). These are the only valid substitutions, though (for example, expressions of the form "y < x > z" are *not* valid). Failure to follow these rules when creating relational expressions of this kind will likely result in unintended calculation results.

Logical operators

Three logical operators are also available: logical AND, logical (inclusive) OR, and logical NOT. AND is represented by the ampersand (&), OR by the vertical bar (|), and NOT by the exclamation mark (!).

Like the relational operators (which return 1 if a comparison is true and 0 if not), these logical operators are zero-centric. That is, any operand that is *non-zero* is treated as true, and *only* a zero operand is treated as false. The returned value is 1 if the evaluation is true, and 0 if false.

Operator precedence

In most situations, you will not need to know the details of which operators bind most tightly. However, when formulas do not appear to calculate correctly, you should check this section to see if precedence is a factor.

Operator precedence is how TreeAge Pro decides where you intended to put parentheses. Consider the following example:

```
A + B * C + D
```

A quick check of the precedence list on the next page indicates that multiplication has higher precedence (binds more tightly) than addition. TreeAge Pro will therefore interpret your expression as:

```
A + (B * C) + D
```

This process is continued until all uncertain bindings are resolved.

The table below lists the operators available in TreeAge Pro in order of precedence. Operators with higher precedence will bind more tightly. Adjacent operators having the same precedence value will be applied from left to right.

Operator	Character	Precedence Value
Unary minus (e.g., "-30,000")	-	8
Logical "Not"	!	8
Exponentiation	^	7
Multiplication	*	6
Division	/	6
Addition	+	5
Subtraction	-	5
Comparators	<, <=, >=, >, =, <>	4
Logical "And"	&	3
Logical "Or"	|	2
Parentheses	( )	1
Brackets	[ ]	1
Function argument separator	;	1

Notice that parentheses are at the bottom of the list. This simply means that the operators inside the parentheses will bind tightly to stay within the parentheses. They are your most useful tool for indicating your particular precedence requirements.

Using mathematical, statistical, and other functions

- TreeAge Pro's built-in functions are case-insensitive, with the name followed by parentheses, and in most cases take arguments in the parentheses.
- Functions that have multiple arguments must use semicolon (";") separators.
- In most cases, any valid expression can be used as an function argument (with the exception of MatrixMult).
- A separate section at the end of the chapter covers the special set of functions that take text string arguments, instead of numeric or variable expressions.
- Functions which here indicate an argument named "LIST" take a flexible number of arguments. For example, Average( ) returns the arithmetic mean of all of its arguments, so Average(1;4;8;13) = 6.5.
- In the tree window, use the Function Helper (see Chapter 13) for help in assigning the correct parameters to each functions.

The functions are described in more detail in the following locations in the manual.

Arithmetic functions
Financial/discounting functions
Miscellaneous functions

See Chapter 14, Building Formulas Using Variables and Functions.

Distribution functions

See Chapter 16, Using Distributions and Monte Carlo Simulation.

Probability functions

See Chapter 35, Markov Modeling Tools and Techniques.

User-defined functions
Global function
List/matrix functions
Node function
String-argument functions

See Chapter 21, Tools for Large and Complex Trees.

Appendix E

Creating Scripts and Interfaces

This appendix briefly outlines a variety of options for working with trees via scripting and custom interfaces, both within TreeAge Pro as well as from other applications.

In this appendix:

- Using the TreeAge Pro Excel/COM module library 580
- Developing TreeAge Pro Interactive applications 581
- Using legacy TreeAgeScript/DATAScript..... 585
- Creating run-time custom interfaces 587

Using the TreeAge Pro Excel/COM module's "TreeAgeProLib" library

Starting with TreeAge Pro 2007, the Excel/COM module (also included in TreeAge Pro Suite) makes it possible for users to create custom Excel macros that can update and analyze trees and other documents in TreeAge Pro. Other Windows applications with COM/scripting interfaces can also use the TreeAgeProLib, for example Visual Basic or C++.

Utilizing your choice of Windows scripting or development tools, with Excel VBA being the most popular, you can control trees, graphs, and Monte Carlo output using a set of commands similar to those used with TreeAge Pro Interactive (see below).

A simple macro might, for example, open the tree in TreeAge Pro, make changes to selected values (probabilities, payoff components, etc.), and then perform a series of Monte Carlo simulations or Markov analyses. The scripting interface can be used to automate repetitive tasks. (The new TreeAgeProLib in TreeAge Pro 2007 includes much the same functionality as the TreeAge Pro Interactive library, called "DATA_Interactive," which is discussed in the next section.)

See Chapter 30, in the Excel Module documentation section, for more details on scripting with the TreeAgeProLib functionality.

Developing TreeAge Pro Interactive™ applications

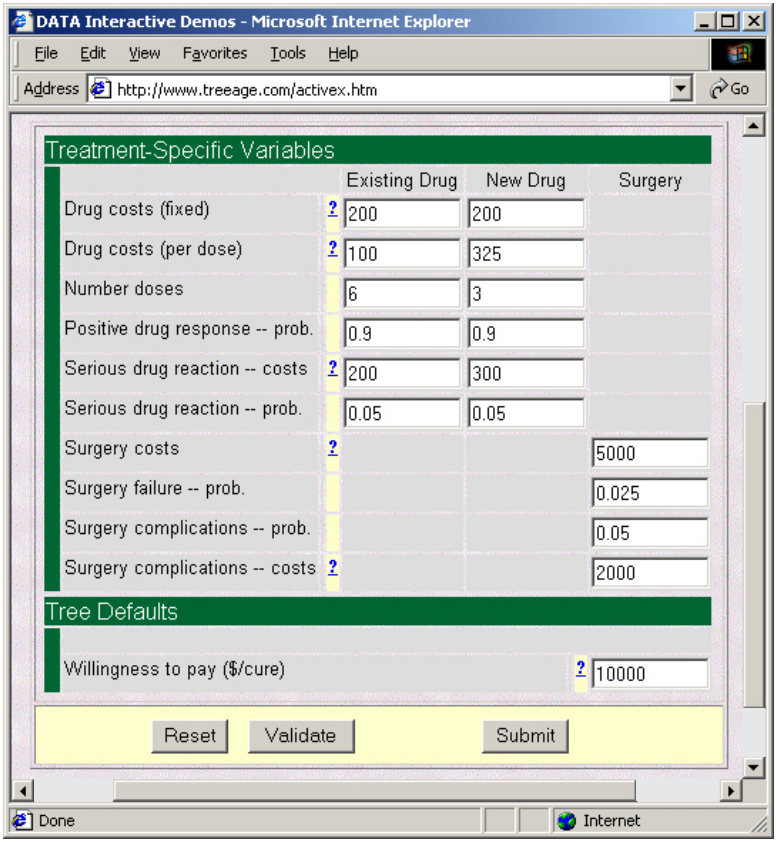
TreeAge Pro Interactive is an ActiveX automation library used to integrate TreeAge models trees with web sites, spreadsheet macros, custom VB applications, and more. As with the TreeAge Pro Excel/COM module covered in Chapter 30, the Interactive version makes it possible for Windows applications to view, update, and analyze trees.

You design a user-friendly interface utilizing your choice of scripting or development tools, and then link it to a decision tree using the "DATA_Interactive" library's straightforward set of commands. A simple interface might, for example, enable remote users to view the tree, make changes to selected values (probabilities, payoff components, etc.), and then perform appropriate analyses, such as sensitivity analysis.

With applications using the "DATA_Interactive" library, it is possible for decision makers without experience using TreeAge to perform sophisticated decision analysis. Your decision analysis application can be accessed remotely, over an intranet or the Internet, or distributed on CD-ROM (or other media) to a targeted audience.

TreeAge Pro Interactive makes it practical to pursue many new applications of decision analysis, for example:

- **R&D portfolio analysis** – facilitates continued monitoring and analysis of R&D projects and their associated uncertainties



- **Dynamic treatment protocols** – enable input of patient-specific information by the treating physician or dispensing pharmacist, allowing decision analyses at points of care

Technical information on the DATA_Interactive library

Web-based demonstrations, as well as trial versions of the TreeAge Pro Interactive software for modelers and interface builders, can be found at the TreeAge web site: http://www.treeage.com/

TreeAge Pro Interactive is a separately licensed software component available from TreeAge Software which allows trees to be integrated with everything from simple VB scripts running on your PC, to stand-alone Windows applications, to complex web sites. In a web environment, since all processing can be done on the *server* side, there is no restriction on the *client* operating system or browser.

The Interactive component provides access to your decision trees via a range of straightforward functions/script commands. These functions are completely documented in the online documentation at:

http://servertreeage.com/objDocs/

The website documents both the DATA_Interactive library as well as the TreeAgeProLib (included with TreeAge Pro's Excel/COM module).

A partial listing of the objects and functions available is provided here to illustrate the power of the component.

TreeAge Pro Interactive – Primary objects

TreeObj	Main wrapper around a decision tree
NodeObj	Represents a single node in a tree

TreeAge Pro Interactive – Selected analysis objects

SensAnParams	Parameters for a sensitivity analysis
SensAnOutput	Holds output of a sensitivity analysis
CEOutput	Holds the output of a baseline C/E analysis
MarkovParams	Parameters for a Markov analysis
MarkovOutput	Holds output of a Markov analysis
MonteParams	Parameters for a Monte Carlo simulation
MonteOutput	Holds output of a Monte Carlo simulation

TreeAge Pro Interactive – Selected output objects

GraphObj	Represents any graphical output
TextReportObj	Provide access to output of analyses

TreeAge Pro Interactive – Selected utility objects

TicketObj	License object required by run-time users
HTMLTableParams	Allows customized output of HTML tables
NumFormObj	Used to set numeric formatting
AxisInfo	Used to set axis ranges

FontObj	Allows setting of fonts
VarInfo	For editing variable definitions at nodes
TableObj	For editing packaged tables

Each object has numerous useful properties and functions. Here is a partial listing of the properties and functions of just the TreeObj object.

TreeObj – File operations

OpenFile	Opens a tree file
OpenPackage	Opens a packaged tree file
IsValid	Ensures that the tree is usable
SaveGraphic	Saves a graphic of the tree

TreeObj – Node operations

SelectRoot	Selects the root node of the tree
SelectRelativeNode	Selects a node related to the current selection
SelectNodeObj	Selects the node from a NodeObj object
GetNodeObj	Creates a NodeObj at the selected node

TreeObj – Selected analysis methods

RollBack	Turns roll-back on or off
Sensitivity	Runs a sensitivity analysis
Markov	Runs a Markov analysis
MonteCarlo	Runs a Monte Carlo simulation
CostEff	Runs a baseline cost-effectiveness analysis
ProbDist	Calculates the risk profile

TreeObj – Selected miscellaneous methods

SetCalcMethod	Sets the active calculation method
SetNumForm	Sets the numeric formatting for the tree
GetTable	Retrieves a TableObj

TreeObj – Selected properties

fileName	The full path of the tree file
selection	Retrieves the name of the currently-selected node
message	Retrieves a status message (for debugging)
lastError	The error code of the last analysis
calcTime	The time required for the last analysis

calcMethod	Retrieves the current calculation method
currentPayoff	Retrieves the current payoff (for Simple calculations)

A sample script using some objects and their functions is shown below to illustrate the usage of the above listed objects and commands. This section of Visual Basic code is based on the macros used in the example spreadsheet "Excel Sample.xls" which is included with TreeAge Pro Interactive. It illustrates how macros use TreeAge Pro Interactive to open a tree, run an analysis, and display results.

```
'------------------------------------------------------------
Public gTreeOpen, gTree   '--- Global variables
'--- This function ensures that all initialization is done.
Public Function MasterInit() As Boolean
    If gTreeOpen = 0 Then
        ok = gTree.OpenFile (ActiveWorkbook.Path & "\simpleCE.tre")
        If ok = 1 Then
            gTreeOpen = 1
        Else
            Msg = "Unable to open tree."
            MsgBox msg, vbOKOnly, "Interactive - Error"
        End If
    End If
    MasterInit = gTreeOpen
End Function

'--- This macro linked to "Baseline analysis" button in spreadsheet.
Sub An_Baseline()
    Dim nObj As NodeObj, ceo As CEOutput , gObj As GraphObj
    If MasterInit() Then  '--- if tree open ...
        gTree.SelectRoot
        Set nObj = gTree.GetNodeObj
        If nObj.type = nt_decision then  '--- if root is decision
            Set ceo = gTree.CostEff
            Set gObj = ceo.GetGraph(1, "")
            '--- calls a macro not shown here
            '    that saves/displays graphic and text report
            ShowGraph gObj, True, 0
        End If
    End If
```

```
End Sub
'-------------------------------------
----------------------
```

Using Legacy TreeAgeScript/DATAScript

TreeAge Pro includes a simple built-in scripting language to automate repetitive tree-building tasks. (Note: TreeAgeScript is not the same as TreeAge Interactive -- see Chapter 25).

Script commands can be run in a number of ways, including from saved text files, as a step in a stored analysis sequence, or from a Script() function in a calculation. A detailed TreeAgeScript Help file is included in the TreeAge Pro installation. A link is included on the Contents page of the regular Help file.

Contact TreeAge Software <support@treeage.com> for more information about particular commands.

> ***Tip:***
> Earlier sections in this appendix, and Chapter 30, describe different options for scripting using the Excel/COM module's "TreeAge Pro" library (or TreeAge Pro Interactive) with Excel Visual Basic macros and or other scripting tools and development environments.

(**NOTE:** It is recommended that you save backup copies of your files before modifying them with script commands.)

Syntax

All script commands in TreeAge Pro begin with an exclamation point, such as "!OpenFile". Commands are not case-sensitive. Many commands take arguments. Arguments must be separated from the command by spaces or parentheses (parentheses are preferred). Each argument may be enclosed in double-quotation marks to ensure that no text is lost. When the argument contains special characters (such as "#" or others described below), use quotation marks. Arguments are separated by either commas or semicolons.

Many commands assign values to script-level variables (as opposed to tree variables). You may also call !SetScriptVar to create a script-level variable. Script variable names are not case-sensitive. Variables are "expanded" in the script (i.e., the name is replaced by the value) when preceded by the "@" sign. When a variable name is required as an argument in a script command, do *not* precede it with "@" (unless your variable value is actually the name of another variable!).

When using file names in commands, use the forward slash ("/") to separate directories, rather than the backslash ("\"). For example: "!OpenFile c:/data/bxr.tre " Also, enclose file names in quotes if the file or any directory in the specified path uses a long name.

Several commands require a node as an argument. Nodes are referenced by their path in the tree. At each branch point, the top node is "A", the second node is "B", and so on. The root node is described by the empty string. The node path "AB" indicates a node which is the second descendant of the first descendant of the root node. These commands allow you to indicate the parent of the selected node using a period. So ".B" will select the second descendant of the selected node's parent, which

will be a sibling of the selected node. You may also use '<' to indicate the node immediately above the selected node, and '>', the node immediately below.

Script command reference

A detailed TreeAgeScript Help file is included in the TreeAge Pro installation. A link is included on the Contents page of the regular Help file.

Parameters in [brackets] are optional; if an optional parameter is omitted, a reasonable default (described for each command) is used. Parameters named "ScriptVar" are script-level variables (not tree variables). Anytime there is ambiguity, the parameters will be more clearly named.

File/Window Commands

!OpenFile (file1 [, file2] [, file3])
!SaveFile / !SaveFileAs (filename) / !SaveAsTRB (filename)
!CloseTopWindow ([doPrompt = false])
!PrintFile
!TopWindowType (ScriptVar)
!TopWindowName (ScriptVar [, doFullName = false])
!CopyPicture

Script-Variable Commands

!SetScriptVar (ScriptVar, value)
!CatScriptVar (ScriptVar, textToAppend)
!CopyScriptVar (ScriptVar)
!ClipAdviseVars (ScriptVar)
!PasteIntoScriptVar (ScriptVar)
!DisplayScriptVar (ScriptVar)
!FlushScriptVars

Flow-Control Commands

!Goto (label)
!Gosub (label) / !Return
!BranchVar (ScriptVar, branch1 [, branch2 ...])
!YesNo (ScriptVar, message)
!BeginBranch / !Branch / !EndBranch

Node Selection Commands

!SelectNode (path) / !SelectNodeToo (path)
!SelectNodeRel (path)
!SelectNodeRelName (nodeName)
!SelectRoot
!SelectIf (category, flags, data)
!DeselectAll
!GetSelCount (ScriptVar)
!GetSelNum (ScriptVar , whichNodesPath)
!GetSelList (ScriptVar)

Node-Editing Commands

!GetNodeName (ScriptVar)
!SetNodeName (name)
!GetNodeProb (ScriptVar)
!SetNodeProb (value)
!GetNodeVar (varName, ScriptVar)
!SetNodeVar (varName, value)
!CopyNodeVars (ScriptVar)
!GetNodePayoff (which, ScriptVar)
!SetNodePayoff (which, value)
!ChangeNodeType (newType)
!AddBranches (number)
!NumKids (ScriptVar [, countEndNodes = false])

Tree-Editing & Analysis Commands

!CopyDocVariables ([showDefaults = false, showCmt = false])
!SetCalcMethod (method)
!SetActivePayoff (which)
!SetNFSigDigs (number)
!TableVal (tableName, index, value)
!RollBack ([onOrOff])
!ExpectedValue (ScriptVar [, doFormat = false])
!PathProb (ScriptVar [, doFormat = false])
!StdDev (ScriptVar [, doFormat = false])
!Evaluate (expression, ScriptVar [, doFormat = false])
!CostEff ([textReportTo])
!ProbDist

Miscellaneous Commands

!Announce (message)

!Delay (milliseconds)
!Test
!TextBox (message [, milliseconds = 3000])

Creating run-time custom interfaces

The Run-time version of TreeAge Pro offers another means of distributing decision analyses to clients (or others in your organization). It has all of the analysis capabilities of TreeAge Pro, but no model-building features. Using TreeAge Pro's built-in Custom Interface designer, in a matter of minutes a simple interface can be set up to allow users of the Run-time software to interact with your model. Low-cost licenses of the Run-time software can be purchased from TreeAge Software, Inc. and downloaded by the end user**.**

Run-time custom interfaces are useful for projects which do not require either TreeAge Pro Interactive's Internet/intranet capabilities, or the high level of control over interface customization possible with Visual Basic and other programming tools. All customization is done in TreeAge Pro and stored in the tree. Simplified input windows allow for changing selected parameters and performing preset analyses.

In addition, users with both TreeAge Pro and a developer license of TreeAge Pro Interactive have the ability to convert a Basic Custom Interface into a more customizable TreeAge Pro Interactive application; refer to the TreeAge Pro Interactive documentation for details.

How it works

The first step, of course, is building the necessary model or models in TreeAge Pro. Designing a Custom Interface for one of your trees is quite simple:

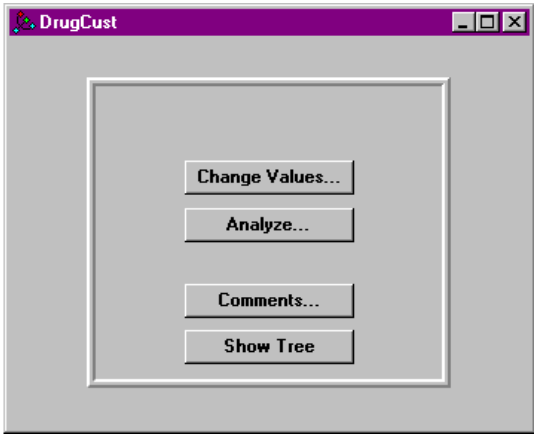
- Before adding the Custom Interface, save a backup copy of the tree.
- Set the display and analysis privileges users should have.
- Decide which variables in your model can be modified by clients. For each variable, you will enter certain information specifically for use in the Custom Interface.
- If you are designing an Extended mode interface, turn on the display of variables in the tree under Edit > Preferences…. Run-time users click on the variables display boxes to modify model parameters.
- Run and store the analyses that you want clients to perform. Stored analyses (listed in the **Analysis >** Storage submenu) will be available for recall by clients.
- Save your tree; the custom interface will be saved with it.

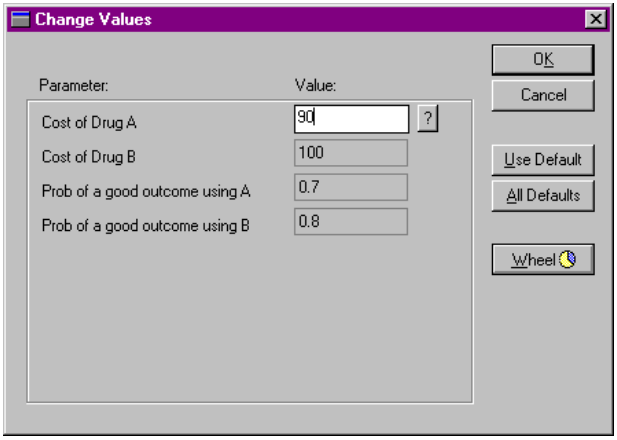
The model builder chooses between two interface modes, Basic and Extended, when designing a tree's Custom Interface. Generally, the choice between the two modes should be based on the

sophistication and needs of the end user. If the interface will be converted into a TreeAge Pro Interactive application, however, then a Basic Custom Interface should be used.

The Basic mode is primarily for clients who do not need (or should not be allowed) to see the tree directly. You may optionally permit them to view the tree structure, but all menu items will be disabled.

When a Basic mode tree is opened with the run-time version of TreeAge Pro, a gray window (the "Basic Client Window" or "BCW") with several buttons will appear. By clicking on the **Change** Values button, a client can change specified parameters in your model by entering the new values in a simple list, without having to select the appropriate node and then open the Define Variable window. The **Analyze** button will bring up a list of analyses prepared (i.e., stored) by the model builder for use by the client.





The Extended mode is for clients who are more savvy about decision trees. When an Extended mode file is opened with Run-time TreeAge Pro, the tree is in full view. In contrast to the Basic mode, where a single Change Values dialog applies to the entire tree, each node in the tree can have its own Change Values dialog box for changing model parameters locally.

In either mode, stored analyses created by the model builder may be run by the client. In the Extended mode, it is also possible to allow clients to run certain of their own analyses, since the

model builder can decide which Analysis menu items are to be disabled and which are to remain active.

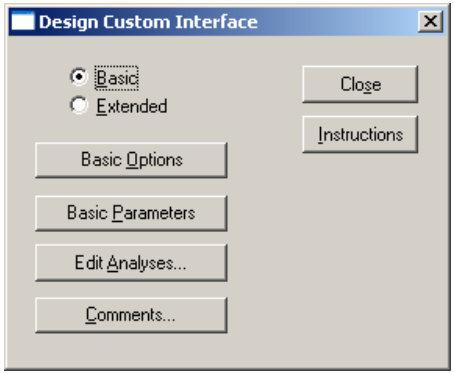
Creating a Basic Custom Interface

The Basic mode has a single Change Values dialog. It applies only to numeric values stored as defaults for the entire tree. Thus, any parameters that will be modified by the client must be stored by the model builder as numeric default values at the root node. In designing your model, remember that the client will not be able to modify any definitions at internal nodes or any non-numeric definitions. Thus, if you are modifying an existing tree for use as a run-time tree, some additional variables, as well as some modifications to existing variable definitions, may be required.

Tip:

Remember to maintain backup copies of your original tree, without the Custom Interface features. The Custom Interface becomes a permanent part of a tree file.

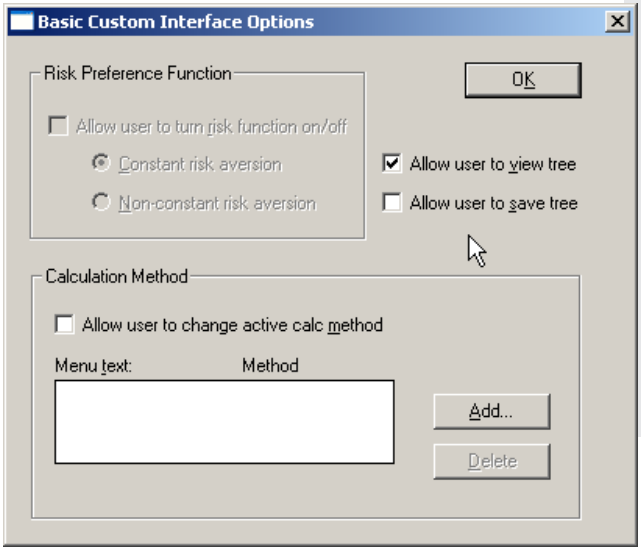
► **To start setting up a Basic Custom Interface:**



▷ Choose **Options >** Design Custom Interface…. Ensure that the Basic radio button is selected and click the **Basic Options** button.

The Basic Custom Interface Options dialog lets you set options for allowing clients to:

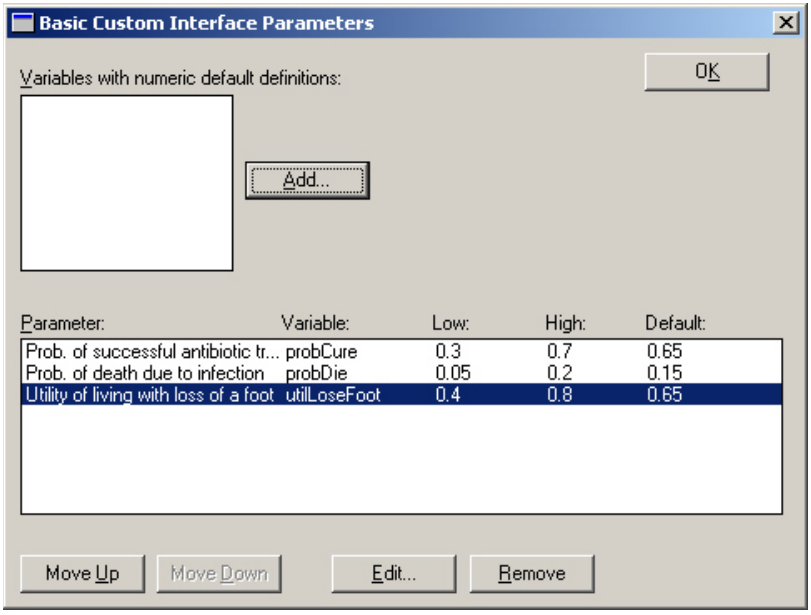
- view the tree;
- save changes to the tree;
- turn a risk preference function on or off (if one has been entered by the model builder); and
- change the payoff being calculated, or the calculation method being used (e.g. from simple to multi-attribute).



If you opt to allow clients to change the active calculation method, they will see a pop-up menu in the interface that will allow them to select from the items you enter in this dialog. The client's current selection for calculation method does not affect the list of stored analyses, so you must indicate for each stored analysis any dependence on a specific calculation method.

To select default values that should be accessible to the client, return to the main Design Custom Interface window and click the Basic Parameters button. The Parameters dialog is shown at the top of the next page.

The list in the top-left corner of the dialog shows all variables with default definitions. At the bottom

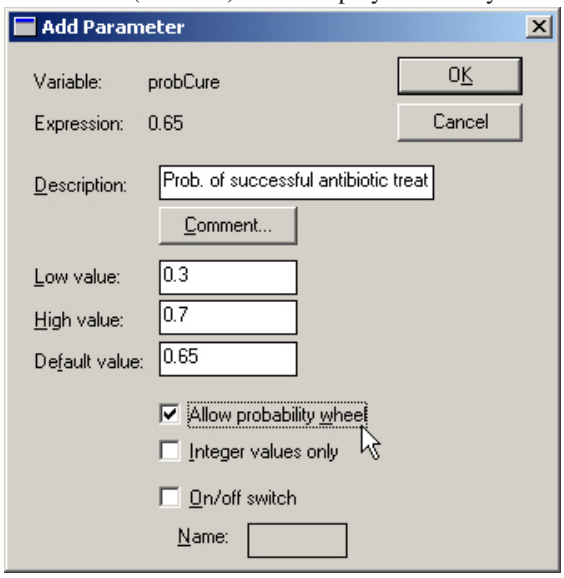


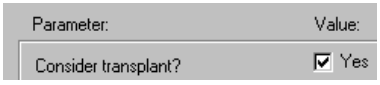
is the list of parameters that clients will see. To move a variable from the top list to the bottom list, use your mouse to select the variable in the top list and click the Add button. This will open the Add Parameters dialog, shown below.

You may change the order of presentation of variables added to the interface display by selecting an item in the bottom list and clicking either the Move Up or Move Down button. If you click Remove, the selected parameter will no longer be included in the

list of modifiable values, and will return to the top list. Clicking Remove will not remove that variable from the underlying tree. Clicking Edit… with a variable selected will open the Add Parameters dialog for that variable.

The description, comment, low value, high value, and default (baseline) value displayed initially in the Add Parameters dialog for a variable are taken from the variable's Properties dialog (see Chapter 13 for details), but can be changed and are maintained separately from the regular set of variable properties.





- **Description**: The text to show the user in the Change Values dialog, instead of the variable's name. If you have previously entered a short description for the variable (in the Properties dialog), it will be used by default.
- **Comment**: If you have entered a comment for the variable (in the Variable Properties dialog), the user will see a button labeled with a "?" to indicate that more information is available for a particular parameter.
- **Low/High values**: These specify the range of values that you will allow clients to enter.
- **Default value**: The client always has the option of *resetting* a given parameter to this default value. Note that the default value is *not* used initially. Whenever a client opens the model, the initial values are those that are saved in the tree.
- **Allow probability wheel**: If the low and high values are 0 and 1, you may allow clients to enter this parameter by using the wheel. Unlike the wheel when it is called directly from the tree window, this option will work only on one value at a time. Thus, if you have a chance node with more than two branches whose probabilities you would like to allow users to change, you have little control over the coherence of the probabilities.
- **Integer values only**: If this box is selected, only integers will be allowed. Otherwise, any real number in the specified range is accepted.
- **On/off switch**: If you select this option, clients will not see a numeric editor in the Change Values dialog. Instead, a check box will be displayed with the name you enter. The client's selection will be stored as 1 if the check box is selected, and 0 if it is cleared. The default value

must be 0 or 1 to indicate the initial state of the check box.

Testing the Basic Custom Interface

As explained above, the analyses that will appear in the run-time interface are those that you have saved via the **Analysis >** Storage sub-menu. See Chapter 18 for detailed information on storing analyses. Note that your basic custom interface will not become active until you store at least one analysis in the tree; instead, the user will see the tree and menus, as in an extended interface.

Every analysis available in the Analysis menu may be stored for use from the Basic Custom Interface except Graph Risk Preference Function, Show Optimal Path, Verify Probabilities, and Roll Back. If you set the Basic Options so that the user can see the tree, they will have the option of running any of these analyses from the Analysis menu. In order to deactivate the Analysis menu when the tree is displayed, the designer must temporarily switch the tree to extended interface mode and disable the Analysis menu items there.

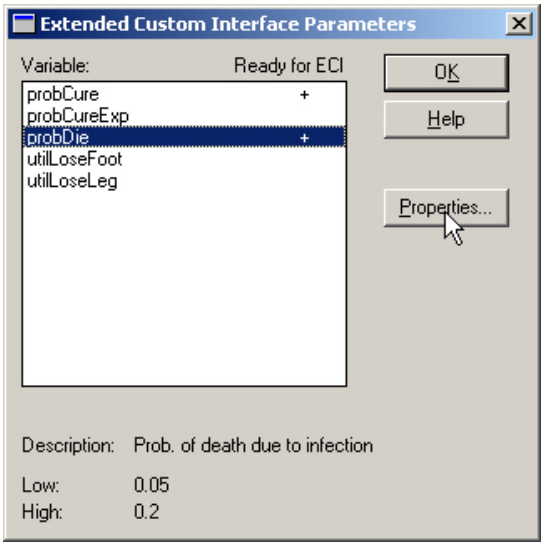
To view the Basic Custom Interface, simply select **Options >** Show Custom Interface. This will activate the BCW. As a designer, you will always have access to the Show Tree button, even if you have not enabled this option for your clients. To fully test

the interface, open the tree using a purchased Run-time version of TreeAge Pro.

Creating an Extended Custom Interface

When you select Extended Options from the Design Custom Interface dialog, you will be able to select which analyses clients will be able to perform manually. Typically, you should prepare most analyses using the Analysis > Storage sub-menu, but you may wish to allow clients to experiment with performing some simpler analyses on their own. In this event, you should consider disabling the sensitivity analyses, threshold analysis, and tornado diagrams, as these require considerable knowledge of TreeAge Pro's interface. This will avoid the risk that your client will get a result that has the appearance of being correct, but is not.

From the Design Custom Interface dialog, press the Extended radio button to begin designing the interface. Press the Extended Parameters button, and you will see a list of all variables in your tree.



A variable will be ready for use in the Extended Custom Interface if all of the following apply:

- it is shown in the tree;
- it has an assigned range of values (the low and high values will be used to check the validity of the user's input);
- it has a short description; and
- it is *not* a Monte Carlo tracker variable.

You may verify these characteristics (or set them) in the Properties dialog box, which you can access by pressing the Properties button either here or in the Define Values dialog box. If all of the above criteria are met, a "+" symbol will appear in the Ready for ECI column in the list box of the Extended Custom Interface Parameters dialog. However, *the*

only definitions of that variable which may be modified by clients are those which are numeric.

If, for example, the variable TestCost is designated Ready for ECI, clients will be able to use the Change Values dialog box (described below) to modify any numeric definition of TestCost. Any instances where TestCost is defined in terms of another variable, or where the definition consists of an expression, will not be modifiable.

These limitations are not necessarily imposed when the user runs analyses directly from the Analysis menu, if you so permit in the Options dialog, discussed above. For example, if you permit the client to select Analysis > Sensitivity Analysis..., the client will be able to run a sensitivity analysis on any tree variable.

When an Extended mode tree is opened by a client using the Run-time version of TreeAge Pro, the full tree appears. To run analyses previously prepared and stored by the designer, clients will need to choose **Analysis >** Storage > Run. The ensuing dialog will be the same as that shown for the Basic Custom Interface.

In addition, clients may be allowed to perform their own analyses, as described above.

Changing values in the Extended Custom Interface

To open the Change Values dialog box at a particular node, the client must double-click inside the variables box at that node. The variables box is the list of variables that is displayed below the node name when the model builder has elected to show variable definitions in the tree. (To use the Extended Custom Interface, this flag *must* be set; see the Variables Display page of the Preferences dialog.) Not all variables that appear in the variables box will be editable in the Change Values dialog box, nor will all nodes that have a variables box necessarily display a Change Values dialog box. See the section about parameters, above.

Clients will not be able to modify probabilities directly. If you wish to allow clients to change probabilities, you must either set up the appropriate probability variable for use in the Change Values dialog, or enable the probability wheel (discussed below). All probabilities will be displayed as their numeric equivalents, even when roll back is turned off.

The probability wheel can be very useful in the Extended Custom Interface. In order to enable its use at a particular chance node, you must ensure that at least one branch has a probability stated as a variable that has been prepared for the Extended Custom Interface. (This variable need not be shown in the tree.) Clients using Run-time TreeAge Pro will not be able to change the storage location for

probability values. See Chapter 12 for detailed instructions on using the probability wheel.

Testing the Extended Custom Interface

When building a model's Extended Custom Interface, you may want to test it from time to time to see how it will look when opened with Run-time TreeAge Pro. This is possible from within the full version of TreeAge Pro.

To see how a model's Extended Custom Interface will look to someone with the run-time version, simply pull down the Options menu, and choose Mimic Run-Time. If you set this option, TreeAge Pro will enable and disable menu items exactly as it does for users of Run-time TreeAge Pro. To turn off run-time emulation, simply repeat the same menu selection.

Protecting your intellectual property

If your model is going to be distributed to users outside of your organization, you may want to take a few additional steps to help protect the model's intellectual property content.

It is possible to prevent anyone from using the full version of TreeAge Pro to open your model, specifying that a particular model can be opened only with Run-time TreeAge Pro. You may also specify a start-up message (e.g., a license agreement) to be displayed when the model is opened. Any attempt to open a "Run-time-only" tree with the regular TreeAge Pro software will generate a message that the file can be opened only with Run-time TreeAge Pro. This level of protection may be desirable even if you are not concerned with protecting intellectual property, but only with ensuring that users not misuse or make undesirable changes to the model.

For information on how to implement these features, contact TreeAge Software technical support (support@treeage.com).

Appendix F

Technical Notes

In this appendix:

- Changing the storage location for tables 598
- Using TreeAge Pro's command line 599
- Other application options/details 600
- Importing SMLTREE and DecisionMaker trees 601

Changing the storage location for tables

It is possible to change the directory in which global tables (those shared by all trees) are stored. This is particularly useful for network installations where the software is stored on a network drive on which users do not have write-permission. To change the directory, you must create a "pointer" file which specifies the name of the directory or folder to be used for table storage.

Create a plain text file called "TABLE.DIR" in the root application directory (e.g., "C:\Program Files\TreeAge\TreeAge Pro\"). On the first line of the file, type the name of the directory which will be used for table storage. For example, your TABLE.DIR text file might contain the text "C:\Med\Info\Tables" to cause TreeAge Pro to look in that directory for tables. On a network installation of TreeAge Pro, this could be used to give each computer a private tables directory.

If the specified tables directory does not exist when TreeAge Pro is opened, TreeAge Pro will ask if you want to create the directory. Any changes you make to the TABLE.DIR file will only be recognized by TreeAge Pro the next time you start the program.

See Chapter 17 for more on working with table files.

Using TreeAge Pro's command line

A number of parameters can be used on the command line used to start up TreeAge Pro.

Using custom start-up flags

You can employ one or more start-up flags to alter certain functionality:

-m*N* Specifies the number of threads to execute during Monte Carlo simulation; this overrides TreeAge Pro's default behavior, which is to use one thread for each available processor (up to 8) on the computer.

-i Tells TreeAge Pro that you usually prefer to input models as influence diagrams rather than trees. The File > Open and File > New dialogs will default to influence diagrams.

-t Suppresses the display of a new, empty tree.

These flags are invoked on the command line of the executable. To set the command line, create a shortcut to TreeAge Pro, open the shortcut's Properties dialog, and edit the Target field (this shortcut can be renamed to reflect a tree being opened).

Startup files

It is possible to designate the names of files to be opened at startup. The names of these files are specified on the command line, in which case the "-t" flag is automatically applied. If the file name, or any directory in its path, contains spaces or punctuation, you must enclose the file name in quotation marks.

Defining variables

You may automatically define variables in the specified tree upon startup. For example, the target field of a shortcut icon in Windows might read:

"C:\Program Files\TreeAge Pro\TreeAge Pro.EXE" -i c:\MyTree.tre?v1=3?v2=v1^2;v3=x

TreeAge Pro will open MyTree and define three variables at the root node:

v1 = 3 v2 = v1 ^ 2 v3 = x

There must not be any whitespace in the formulas. The question mark is the delimiter.

Influence diagrams

Here is an example of a command line with flags set to open TreeAge Pro with influence diagrams as the default model type, with the startup file MyInf.ind:

"C:\Program Files\TreeAge Pro\TreeAge Pro.EXE" -i "c:\files\MyInf.ind"

Other application options/details

A variety of other, technical items related to the TreeAge Pro interface are described below.

Beginner visual cues

To help new users enter probabilities and node labels on the face of the tree, TreeAge highlights and labels these editing areas when the mouse is in the correct position.

These cues can be turned on/off via the **Help >** Show Beginner Cues command.

Suppressable dialogs

Some user prompts and messsages have been made suppressable; in other words, a checkbox is provided that instructs TreeAge Pro not to show the dialog again. If a dialog that asks a question is suppressed, TreeAge Pro will store and continue to use the last answer provided.

All suppressed dialogs can be re-enabled via the **Help >** Enable Suppressed Dialogs command.

Window sizing

TreeAge Pro will automatically re-maximize the tree window when you close a definition-type dialog/window (e.g., Define Variable, Enter Payoff) if the tree was maximized prior to opening the definition-type dialog/window.

Hot keys when opening a tree

The CONTROL, SHIFT, and ALT keys have special uses while a tree is being opened. Their purpose is to make it easier to work with large models by making a change to the tree before it is displayed.

CONTROL = reset tree's preferences back to current defaults

SHIFT = check for bad variable names (e.g., conflict with table)

CONTRL+SHIFT = reparse all formulas

SHIFT+ALT = collapse the tree at the root node

CONTROL+SHIFT+ALT = suppress clone updating, hide clone copies

Hot keys when opening the Preferences dialog

The CONTROL and SHIFT keys also have special functions when the Preferences dialog is being opened, allowing preferences to be reset.

CONTROL = reset tree's preferences back to program defaults

SHIFT = reset default preferences back to program defaults

Importing SMLTREE and DecisionMaker trees

TreeAge Pro can read most SMLTREE and Decision Maker files directly. Simply use the **File >** Open command to open a SMLTREE file as though it were a TreeAge Pro file. First, however, you must change the file extension to "*.tre" so that TreeAge will recognize the file as a tree.

TreeAge Pro will create a new, untitled decision tree containing the imported file; you should save the imported tree with a name or location different than the original file.

Contact TreeAge technical support (support@treeage.com) for more information on SMLTREE/Decision Maker import diagnostic messages, conversion limitations, common import problems, bindings, tables and more.

Import options

Due to conceptual differences in tree format, the import facility should be regarded as an initial step in a multi-step process. It is possible that some manual modifications to the tree will be required. When you open a tree created by SMLTREE or Decision Maker, you will see the File Import Options dialog. In it, you can specify three preferences:

- **Use clones at duplicated subtrees** – Check this box to duplicate subtrees using the clone feature. See Chapter 21.
- **Use hash mark explicitly in last branch of each chance node** – If you select this check box, the bottom branch of each chance node will be changed to use # for the remainder probability. (Decision Maker effectively ignores the probability on the lowest (lower) branch, so it is possible that your model contains sets of probabilities that do not sum to one.)
- **Truncate identifiers to 8 characters** – SMLTREE and Decision Maker required names to be 8 characters or less, while in a formula you were allowed to use longer names. If, for example, you were to enter a formula *total* = *costSurgery* + *costMedicine*, SMLTREE or Decision Maker would interpret the formula as *total* = *costSurg* + *costMedi*. If you select this check box, TreeAge Pro will convert the former formula to the latter, thereby preventing calculation errors.

If there are multiple unattached nodes in the file, you will be asked to specify which should be used as the root node of the new tree. If you repeat the import process a number of times, you can convert separate trees with each unique root node.

Index

Symbols

! operator 577
(complementary probability) 144
& operator 577
.TRB file 540
? operator 591
_sample
  keyword, Monte Carlo 512
_trial
  keyword, Monte Carlo 512
| operator 577

A

acceptability curve
  interpretation 436
  net benefits 436
ActiveX 581
ActiveX client. *See* TreeAgeProLib
  using the TreeAgeProLib 362
align
  influence diagram nodes 342
analyses
  Markov Cohort 458–464
  Over/Under 70
  Probability Distribution 57
  Rankings 55
  Verify Probabilities 69
analysis sequences 251, 513
  adding stored analyses 251, 252
  adding variable definitions 252
    evaluate at root node 252
    evaluate immediately 252
    non-numeric 252
    numeric 252
  creating 251
  examples 513
  Monte Carlo simulation 216
  removing sequence items 253
  running 253, 254, 514
  sequence stored analyses 251
  simulation using trials
    Markov processes 513
  storing 514
  uses 251
  variables definitions
    resetting pre-sequence values 255
analysis storage
  analysis description 248
  available analyses 248
  deleting stored analyses 249
  maintaining stored analyses 249
  rename stored analyses 249
  running stored analyses 249
  saving last analysis 248
  tornado diagram 189
  using graph templates 86, 250
  with run-time models 247, 256
ApplicationObj. *See* TreeAgeProLib: object documentation
arcs 8, 13
  annotating 341
  curving 341
  editing 341
  flipping 341, 348
  fonts 572
  grayed out 333, 572
  influence 13
    probabilistic 13
  on clipboard 341
  structure-only 572
  timing 332, 344, 346
arrow keys 538
arrows 568
  aligning 120
  creating 120
asymmetry 36, 333
  default variable values 336–342
  elim 333
  force 333
  skip 333
  skip all 333
  symm 333
autosave 574

B

backup and autosave 531, 574
backward-compatibility 3
batched analysis 251
Bayes' revision
  binary hypothesis 312
  dichotomous test 312
  example 310, 312
  influence diagrams 344
  introduction 310–311
  multiple tests 330, 347
  performing 313
  preparing the subtree 312
  sequential tests 310
  structural requirements 312
  tree window 310
  using the dialog 313
Benefit-Cost 89. *See also* calculation method: multi-attribute: benefit-cost
Beta distribution (integer form) 225
bi-directional links 268. *See* dynamic links; *See also* dynamic links; *See also* Excel add-in
bitmap 573
branches. *See* decision tree: branches
budget impact
  using dynamic Markov cohort 495

C

calculation method
  attributes 88, 89
  benefit-cost 94
  custom, creating 91
  multi-attribute, generalized
    setting weightings 93
  optimal path
    high or low 559
  status bar indicator 88
calculation preferences. *See* preferences: calculation method
Calculator/Evaluator tool 158
certainty equivalents 9
clipboard 108
clones
  influence diagram 349
  suppress updating
    before opening tree 600
clones (tree)
  clone copies
    attaching 283
    detaching singly 284
    hiding 286
    root node type 283
  clone master 282
    creating 282
    modifying properties 283
    naming 282
  Clones dialog 284
  explorer view display 274
  hiding 566
  introduction 282
  modeling strategies 286

nesting 285
suppressing clone updating analysis 287
closing files 287
reenabling clones 287
using variables 285
vs. pasted subtrees 282
Clones list dialog 283
COM automation. *See* Excel module: VBA macros; *See also* TreeAgeProLib
command line 599
comments
node 564
complementary probability 144
concatenation
string function arguments 290–297
cost-effectiveness 21, 25. *See* Healthcare Module: cost-effectiveness
calculation method 395–397
decision-making 408
graph options 82
influence diagrams 330
inverted effectiveness 412, 413, 416, 417
in Markov calculations 415
inverted incremental effectiveness 412, 416, 417
optimal path criteria 409
evalutating variables used in 416
maximum cost 416
minimum effectiveness 416
setting 416
using variables in 416
payoff assignments 394
preferences 395, 396
willingness-to-pay 416, 416–417, 432, 436
use in net health benefits 416–417
cost-effectiveness analysis 398
absolute dominance 398
dominance 398, 400
expected values 399
extended dominance 398, 400
ICER 399
incremental values
cost 399
cost-effectiveness 399
effectiveness 399
interpreting zero values 423
inverted effectiveness 396
inverted incremental effectiveness 396
least costly option 399
net health benefits 408
advantages vs. CEA 408
formula 408, 418–424
optimal path selection
algorithm 409
embedded decision nodes 417
using calculated ICERs 417
using costs 417
prerequisites 394–397
Rankings 399
roll back 425
cost-effectiveness graph 398, 406
axis options 398
dominance
inferring by position 400
extended dominance
blending two options 404
representing in graph 404
plotting 398
Cost-Effectiveness in Health and Medicine 392, 433
cost-effectiveness report 399
multiple tables 399
cost-effectiveness sensitivity analysis, 1-way 405–406
animated cost-effectiveness graph 406
ICER line graph 406, 418
asymptotic lines 423
customizing 423
dominance 423
negative ICER options 406
negative ICERs 423
plotting dominated alternatives 423
threshold approximation 418
willingness-to-pay line 418
zero and negative ICERs 422, 423
line graphs 406
average cost 406
average effectiveness 406
incremental cost 406
incremental effectiveness 406
net health benefits 406
net health benefits 406, 408
output window 405
graphing options 405
text report 405
thresholds 418
cost-effectiveness sensitivity analysis, 2-way 424
graph
cost thresholds 424
dominance 424–425
interpreting regions 424
region of optimality 424
isocontours 424
incremental C/E regions 424
order of node branches 424
vs. cost or effect only 424
custom distribution table 232
custom interface 587
basic mode 587, 588
calculation method 589
risk preferences 589
testing 592
change values 588, 594
extended mode 588, 593
ready for ECI 593
run analyses manually 593
user view 594
protecting intellectual property 595
stored analyses 256, 588, 592

D

DATA_Interactive library. *See* TreeAgeProLib
object documentation 582–585
DATA 3.5
changes from 531–534
DATA file compatibility 3
DEALE 474
Debugging
variable calculations 158
debugging 272
microsimulations 295, 507, 569
debugging preferences 295, 507
Debug pane 158
debug pane 53
Debug( ) function 295, 507
Decision Maker
bindings 601
importing trees 601
variables 601
decision probabilities 310
decision tree
adding branches
default number 566
alternatives 10
basic calculations 12
basics 10

branches
adding 18, 104
changing order 107
deleting 18, 106
inserting 105
inserting sets 105
reordering 107
building 16–17
calculation method 559
calculation preferences 23
chance nodes 10
branches 10
expected value 12
change node type 20
change payoff 21
dealing with large models 272
decision nodes 10
branches 10
expected value 12
design 10
display
customizing 131
preferences 131
Explorer view. *See* explorer view
label node 116
label nodes
uses 116
linking 286–298
logic nodes 98
minimize empty space 566
moving around. *See* decision tree: navigating
moving around in 19
navigating 17, 19
node comments 117
nodes
calculating expected value 53
deleting 106
deleting end nodes 106
numeric formatting 54
outline format 276
payoffs 20–21, 21
probabilities, entering 22
reducing complexity 272
reducing size 272
root node 17
terminal nodes 10
payoffs 10
uncertainties 10
default settings 558
deterministic sensitivity analysis 9
Dirichlet distribution 302, 499–500, 536
Discounting
formula 473
UtilDiscount() function 473
discounting 178
discrete-sized "population" 301
discrete event models
distribution sampling rate 221
display 103, 113–114
DistKids function 304
distribute children
manual correction 307
Distribute Children command 304
distributions 161, 224–231, 234–244
advanced properties 206, 221–223
Sample once per Markov stage per trial 207
approximating parameters 223
as parameters 222
correlating 222
Numerical Python 222
custom
sampling in Excel 244
table, data set 231
deleting 208
DistSamp() function 206
Excel charts 365
first-order (individual variability) 512
formulas and parameters
beta 225
binomial 229
chi 225
chi-squared 226
Dirichlet 225. *See also* distributions: multi-variate
Erlang 226
exponential 226
gamma 227
hyper-exponential 227
Laplace 227
log-normal 228
logistic 227
Maxwell 229
normal 224
Poisson 229
Rayleigh 229
triangular 224
uniform (linear) 224, 225
Weibull 229
importing lists 208
in influence diagrams 339
managing lists 208
modifying 208
multi-variate
Dirichlet 225
table, custom. *See* distributions: custom: table, data set
numeric indexes 205
palette 206
parameterizing 205
defining parameters at root node 222
using non-numeric parameters 222
properties
comment 206
index 206
Properties dialog 206
representing second-order uncertainty 512
resampling per Markov stage 221–231
sampling during EV calculations 222
sampling during expected value calculations 514
sampling from tables 230
table, custom 232
uses 205
docking
explorer and properties views 16
dynamic links 356–362, 372–381. *See also* Excel
calculations 262
Dynamic Links dialog 360–361
influence diagram 372–380
outputs to TreeAge 356, 357, 359
spreadsheet recalculations 262
trigger link (DDE) 259, 260, 261
dynamic treatment protocols 582

E

error message
missing Markov reward 452
error messages 69
European/regional numeric setting 394, 450
example models 2
examples
airline problem 188
drug trial Markov 468, 483
faulty machine 344
investment decision 10
sensitivity analysis 148
Markov bindings 480
Markov Monte Carlo #2
probabilistic sensitivity analysis 509

Markov tunnel 486
oil drilling 68
prostate cancer 333
rich uncle 320
rock climber treatment 191, 198
three state Markov 448–468
Excel add-in 353-385
TreeCalc( ) worksheet function 361
VBA macros 362
Excel macros. *See also* TreeAgeProLib
Excel module 73, 80. *See also* Excel add-in
tables
adding and editing 369
variables
editing and adding 367
VBA macros 361, 362
Excel spreadsheet 258
converting to influence diagram 372
linking 357. *See also* dynamic linking
expected utilities 9
expected value 12
calculation 12
single node 53
expected value of perfect information (EVPI) 65–70, 330, 341
defined 65
in influence diagrams 348
using Monte Carlo simulation 204
explorer view
tree 272
copying/cutting subtrees 275
export 573
exporting
images. *See* exporting images
exporting images
formats
.TRB files 47
bitmap 45, 46
GIF 46
JPEG 46
metafile 45, 46
saving graphics files 44
screen capture programs 47
to graphics files 44
to PowerPoint 46
via the clipboard 45
subtrees 45
extended dominance 401

F

file compatibility 3
find text
formula text 113
node text 113
variable definition text 113
folding back 12
fonts 74, 132
arcs 572
default for nodes 567
graph labels 74
graph template 85
influence diagram 572
node 132
node comment 117
note boxes 118
subtree 133
TrueType 573
functions 168, 576–578
argument separator (;) 169
arithmetic 176–183, 578
Abs 177
Average 177
Ceiling 177
Exp 177
Floor 177
Int 177
Ln 177
Log 177
Max 177
Min 177
Modulo 177
Prod 178, 578
Root 177
Round 177
Sqrt 177
Sum 177
BilinkName 297
Command
Excel commands 296
Debug 294
Dist 221
DistForce 221
Excel bilinks 297
Excel macro calls 294
ExportGlobalMatrix command 294, 296
financial 475–500
Annuity 178
Compound 178
FV 178
NPV 178
PMT 178
PV 178
Function Helper 169
Global 296
Global() 303, 497, 499
GlobalN 296
helper 176, 578
List 297
Markov-related 474
Matrix 297
miscellaneous 179–183
Choose 179
dist 220
If 179
Inf 179, 474
Pi 179
Stdev 177
Node( ) 290
using extra payoffs 91
probabilities/rates
DEALE 474
OddsToProb 474
ProbToRate 474
RateToProb 474
UtilDiscount 178
Python 293
PythonGetList 293
PythonRun 293
PythonRunFile 293
Script 294, 295, 507
StateProb 474, 496
string/text 292, 293
syntax 176, 578
Tree( ) 292, 293
user-defined, Python 288

G

getting started
example models 2
web resources 2
GIF images. *See* exporting images: formats: GIF
Global() function 296
Global function 497. *See* functions: Global()
Gold, Marthe 392
GraphObj. *See* TreeAgeProLib: object documentation; *See also* DATA_Interactive library

graphs
alpha markers 532
customizing 72
custom labels 75
graph template 85
custom lines 78
graph template 85
labels 73
editing text 74
fonts 74
line colors
changing 82, 83
markers
changing 82, 83
numeric formatting 77
Options dialog 76
templates. *See* graph templates
contents 85
creating 85
modifying 86
text reports. *See* text reports
types
bar 81
line 81
region 84
scatter plot 85
graph templates 85. *See also* analysis storage
analysis storage 86

H

Healthcare Module
cost-effectiveness 89
Healthcare module 397
hide node texts 128, 564
hiding probabilities 128, 564
histograms 81. *See* probability distribution
hot keys
trees, when opening 600

I

imperfect information 68
incremental cost-effectiveness ratios 399
incremental cost-effectiveness ratios (ICERs) 401
individual variability
representing with distributions 217, 221. *See also* microsimulation: distribution sampling
infectious disease model. *See* Markov microsimulation: parallel trials; *See also* Markov models: dynamic cohort
influence diagram 13
advantages 13
analyzing 14
annotating 340
arcs
adding 30
deleting 31
gray 37
selecting 30
Bayes' revision 344
Bayes' revision
decision node required 346
multiple tests 347
chance nodes
probabilities, entering 34
clones 349
create 349
destroy 349
collapsing 350
converting to a tree 38
converting to tree 571
algorithm 331
decision node
alternatives, add 32
alternatives, delete 33
alternatives, reorder 33
deterministic nodes 334
distributions
in the converted tree 339
distributions, using 339
EVPI 348
fixed node size 571, 574
graphical position of nodes 342, 571
hiding nodes 350
importing from spreadsheets 372
introduction 8
limitations
cost-effectiveness 330
Markov processes 330
multi-attribute 330
linking to spreadsheets 339
Markov processes 330
mini-tree 337
multi-attribute 330
node comments 340
nodes 13
creating 29
selecting 29
note boxes 340
payoff formula 335
redraw problems. *See* Redraw Window
sub-models 350
value nodes
assigning a formula 335
enumerating 335
payoffs, entering 35
when to use 14, 330
influence diagram window 28
information 9
inverted effectiveness. *See* cost-effectiveness: inverted effectiveness

J

JPEG images. *See* exporting images: formats: JPEG

L

label node 116
life expectancy 450
likelihood probabilities 310, 312
linking 258–269. *See also* dynamic links
linking (client-side) 264, 265–269, 298
capturing server data 265
cell ranges 262
copying links 268
invoking bi-directional link 268
link properties 265
assign index 267
path 268
replace path 268
Link Properties dialog 264
Links dialog
invokes bi-link 267
link OK 267
requirements 266
spreadsheet
broken links 266
cell formatting 266
cell ranges 262
Sub() function 264, 265
to spreadsheet
named cells 266
worksheet order 266
linking (server-side) 264, 269
Copy Special 269
drawbacks 269

logic nodes
Markov microsimulation 516
LOGNORM4
Lognormal estimator software 228
lottery 9

M

macros. *See* Excel module: VBA macros
Markov
jump states
find/replace 535
transition bindings
find/replace 535
transitions
find/replace 535
Markov cohort analysis 458, 458–461
automation/scripting 500. *See also* TreeAgeProLib: MarkovOutput
cost-effectiveness 465, 466
Excel charts 365
graphs 461
multiple Markov processes 465, 466
output window 459
technical details 519
text report 460
trace 460
Markov microsimulation 501–516
decisions in Markov processes 516
decisions using trackers 516
dynamic cohort. *See* Markov microsimulation: parallel trials
examples 504
parallel trials 515
sensitivity analysis 504
sequencing trial-only simulations 513
simulation using samples 508
theory 509, 510
simulation using samples and trials
interpreting output 510
simulation using trials 502–506
termination condition 516
using tracker variables as inputs 504–506
StateProb function. *See* Markov microsimulation: parallel trials
technical details 520
tracker variables
as inputs 504–506
limitations 502
modifications 502
storing history 502
uses 502
using 502–506
using with logic nodes 504
using expected value analysis 504
versus cohort analysis 465
Markov models 446–468
absorbing states 453
analyzing 458–464
final state probabilities 458
using roll back 458
basic elements 448
basic models 446–468
building in DATA 448
calculation basics 447
cohort analysis 447
simulation trials 447
calculations
cohort life expectancy 450
prerequisites 452
clones 480
cloning a state's transitions 480
cloning entire Markov processes 480, 483
entire subtree 483–500
cohort analysis 502
cohort analysis 502
cost-effectiveness models 466
setting C/E termination condition 467
cycle length 448, 450
determining rewards and probabilities 450
cycle tree 448
cycle zero 470
explained 470
first transitions 470
half-cycle correction 470
initialization phase events 470
decisions and logic nodes 516
Dirichlet distribution 499–500
discounting rewards 473
adjusting for cycle length 473
discrete vs. continuous models 446
displaying Markov information below nodes 451, 481
disutilities 478
dynamic cohort. *See* non-coherent probabilities
changing "population" size 494–496
infectious disease model 495
parallel trials 496
StateProb function 496
fixed values 471
half-cycle correction 465, 470, 476–479
automatic (button) 476
in absorbing processes 465
initial reward component 465
incremental rewards 448
vs. time-independent rewards 479
initial distribution 470
initial probabilities 446, 448
entering below states 449
Jump-To dialog 455
keywords 470
_stage 467
_stage_cost 467, 470
_stage_eff 467, 470
_stage_reward 467, 470
_total_cost 467, 470
_total_eff 467, 470
_total_reward 467, 470
_tunnel 470, 484, 487
cost-effectiveness calculations 467
using in termination conditions 470
logic nodes 516
Markov bindings 480
defining 481
referencing in clone master 482
Markov node 448
Markov State Info dialog 466
defining a tunnel state 486
editing reward sets 466
Markov subtree 448
onetime rewards
after cycle zero 478
outline format. *See* decision tree: outline format
prior costs 477
prior utilities and disutilities 477
representation 446
bubble diagram 446, 453
DATA cycle tree 446
rewards 448
accruing from prior to Markov process 477
outline format 281
reward sets 466

stage counter 470
stage reward 470
stages 446
state bindings
  Tunnel Crossover binding 488
  Tunnel Table binding 488
  Tunnel Variable binding 488
state rewards 448, 450–468
  defining 450
  final 450
  incremental 450
  initial 450, 470
states 449
  adding 449
  naming 482
survival curve 461
termination condition 448, 457, 516
  cost-effectiveness 467
  default 457
  during simulation trials 516
  outline format 281
  stage reward threshold 472
  using cost-effectiveness keywords 467
total reward 470
transition bindings
  outline format 280
transition nodes 455
  changing the jump-to state 455
  using terminal nodes 455
transition probabilities 446, 448, 453
transition probability tables 472
  including stage zero values 470
  interpolating values 472
  referencing 472
transition rewards 448, 465, 478
  accounting with jump-to states 479
  accounting with state rewards 479
  when to use 478
transitions 453
transition subtree 453, 486
  creating 454
tunnels
  merging temporary states 487
tunnel states 484
  chain of temporary states 485
  defining 486
  description 484
  outline format 281
  specifying # temporary states 486
  uses 484
  using the _tunnel keyword 487
Markov Monte Carlo methods. *See* Markov microsimulation
  termination preferences 562
MarkovOutput. *See* TreeAgeProLib: object documentation; *See also* DATA_Interactive library
Maximin roll back 96
  calculations 96
  display 96
  optimal path 96
  roll back boxes 96
  use 96
maximizing
  tree window 532
Medical Decision Making 433, 435, 440, 508
memory
  in a Markov model. *See* tracker variables
memory limits
  Windows XP 214
menus 540–555
  Analysis 551
  Diagram 553
  Display 545
  Edit 542
  Graph 554
  Options 549
  Table 555
  Values 547
metafile 573
microsimulation. *See* Markov microsimulation; *See also* Monte Carlo simulation: microsimulation
mini-tree
Minimax analysis 96. *See also* optimal path: reversing at a decision node
modeling
  decision trees 10
  sensitivity analysis 138
Monte Carlo simulation 9, 202–218, 221
  at decision nodes 216
  comment field 216
  decision nodes 216–218
  distributed simulations 214
  downstream decisions 216
  Excel charts 365
  first-order 203
  identifying values
    adding 216
    removing 216
  memory requirements 214
  microsimulation 217. *See also* Markov microsimulation
    distribution sampling 221
  random number generator 215
  sampling options 215
  second-order 203
  seeded simulation
    Distributed consistency 215
    Independently seed 215
    microsimulation variance reduction 215
  setup dialog 209
  three-dimensional simulation 218
  two-, three-dimensional 532
  two-dimensional simulation 218
  vs. roll back 202
MonteOutput. *See* TreeAgeProLib: object documentation; *See also* DATA_Interactive library
multi-attribute calculations 93
  benefit-cost 94
  cost-effectiveness 94
  generalized multi-attribute 94
  preferences
    calculation method 93
  weighting function 94
multi-attribute cost weightings 409
multi-processor calculations. *See* Monte Carlo simulation
  local network 203, 214
  single computer 203, 214

N

Navigation button 538
nested trees
  calculating payoffs 298
  calculating probabilities 298
  subsidiary tree 298
Net Benefits 436–441
  acceptability curve 436
  cost-effectiveness EVPI 441
  EVPI curve 441
net benefits
  roll back 408
net present value discounting 300
node comments
  red flag 118

NodeObj. *See* TreeAgeProLib: object documentation; *See also* DATA_Interactive library
Node Outline window 277
nodes. *See* decision tree: nodes; *See also* influence diagram: nodes
  selecting 100–101
    multiple, unrelated 100
non-coherent probabilities
  dynamic Markov cohort 494
note boxes 119
  arrows 120
    drawing 120
    moving 120
    only in trees 120
  binding 119
  deleting 119
  drawing 118
  fonts 118
  preferences 568

O

ODBC database
  loading table values from 234
  recordsets 239
  SELECT query 239
ODBC data source 238–240
operators, math 576–578
  arithmetic 576–578
  logical 577–578
  precedence 577
  relational 576–578
optimal path 98
  force at node 98
  maximize 90
  minimize 90
  reversing at a decision node. *See also* Minimax analysis
  showing in tree 98
  switching between low and high 98
optimization 9
options valuation 300
outline tree format 276. *See also* decision tree: outline format

P

package files. *See* tables: packaging with tree
Page Headers & Footers dialog
  decision tree 43
parallel trials. *See* Markov microsimulation: parallel trials
path probabilities
  preferences
    show for all paths 95
payoffs 10, 11
  calculating over/under probabilities 70
  calculating range 69
  displaying 563
  Enter Payoff window 94
    cost-effectiveness 394
  node name as 562
  numeric vs. formula 138
perfect predictor 65
points of care 582
portfolio analysis 581
preferences 23
  calculation method 88, 559
    multi-attribute 93
  default 133
  displaying Markov information 452
  fonts 133–134
    default for nodes 133
    default for probabilities 133
    roll back boxes (EV) 133
    variable definitions 133
  global 573–574
  influence diagram
    arcs 572–574
    fonts 572–574
    node size 571–574
  node display
    boxed payoffs 128
  notes & arrows 568–574
    borders 120
  numeric formatting 151
    graph axis 77
    graph template 85
    influence diagrams 571
    setting 559
    units 52
  other calculation 562–566
  printing 43, 565–566, 573
  regional numeric settings
    currency symbol 97
    overriding Windows settings 97
    specifying 97
  resetting to defaults 558, 600
  risk 325, 561
  roll back 95, 129, 560–566
    calculate payoffs only 127
    terminal node columns 95
  setting defaults 133
  terminal node display 562–566
  tree display 566
    align end nodes 131, 566
    minimize empty space 130
  tree font 567
  variables display 564–566
    displaying Markov information 452
Preferences dialog 558–574
  resetting defaults 532
printing 42–43
  alignment 42
  center in page 570
  headers & footers 570
  page breaks 570
  page setup 42
  preview
    headers & footers 43
    image offsetting 42
    image resizing 42
  show headers 570
  use printer for sizing 573
  zoom 42, 570
probabilistic sensitivity analysis 9
probabilities 11, 310
  complement 144
  complementary 562
  display 560
  error checking (coherence) 69
  hiding 564
  numeric vs. formula 138
  path. *See* functions: PathProb( )
  remainder 144
  verifying/testing 69
  visual cues 600
    turn on/off 600
probability coherence 148
probability distribution 57–70, 81, 304
  bars
    calculating height 58
  comparative 60–70
  intervals, x-axis 58
  probability axis
    auto-scaling 81
Probability Distribution command. *See* probability distribution

probability wheel 591, 594
in influence diagram 338
using # 114
using variables 114
properties grid
editing table values 237
Markov information 450
tree 272–275
tree window 535
Python
embedded interpreter 289
user-defined functions 288
outline format 279

Q

queueing models. *See* Markov microsimulation: parallel trials
Quick menu
creating tracker variables 505

R

R&D portfolio analysis 581
rankings 55
real options valuation 300
redo 112. *See also* undo
Redraw Window 545
replace text
all occurrences 113
formula text 113
node text 113
variable definition text 113
where 113
revert to saved 112
risk preferences
certainty equivalents 321–325
coefficient 323
constant risk-aversion 323
assigning 323
non-constant risk-aversion 325
assigning 324
turning on 561
roll back 12, 24, 54
algorithm 12
boxes 325
contents 95
fonts 567
hiding 55
moving 55
calculations
changing displayed value 95
expected value 95
maximin 95
path probability 95
display
terminal node columns 95
preferences 95
turning off 54
roll back columns
incremental values 124
setting up 125, 126
terminal node numbers 127
turning on 95
visual endnodes 124
Roll Back command 54. *See also* roll back
roll back report 24–25
root node. *See* decision tree: root node
Run-time DATA
building custom interfaces 587
Run-time version 587, 595

S

scatterplots 431
scientific notation 52
screen redraw problems. *See* Redraw Window
scripting 541
command reference 585–587
syntax 585–586
scripting, ActiveX
DataATLLib technical specifications 384–386, 581–582
Select If command 101
selecting nodes. *See* nodes: selecting
sensitivity analysis 9
1-way 138, 150–152
graph marker legend 150
interpreting the graph 150–152
threshold, caveats on 152
thresholds 151–152
thresholds, hiding 151
thresholds, linear interpolation 152
thresholds, reducing error 152
Threshold Values legend 150, 151
2-way 191
indifference, regions of 192
isocontours 194
region graph 191
requirements 191
threshold distortion 192
thresholds 192
thresholds, vs. 1-way 192
3-way 194
animation of 2-way graphs 194
third axis 194
correlated variables 198
excluding correlates 198
interpreting graphs 199
deterministic nodes 334
influence diagram 334
introduction 148
limitations 202
manual 159
methods 148–150
on probabilities 148
on probability variables 180
on risk preference coefficient 325
options 198
correlated variables 186
probability coherence 200
parameter uncertainty 138
precautions
multiple definitions 186
non-numeric definitions 186
requirements 138
threshold analysis 195
explanation 195
interpreting results 200
non-linearity 196
tolerance 195
tips 180, 186
tornado diagram 84, 188–190
correlated variables 190
cumulative risk % 190
hiding bars 189
interpreting 188–189
risk % 190
spread 190
spread, squared 190
storing analysis 189, 192, 193
text report 190
thresholds 189
viewing 1-way line graphs 189
uses 138, 150
simulation graphs 430–441
acceptability curve 436–437
willingness-to-pay 436–437
cost-effectiveness
acceptability curve 436–437
cost-effectiveness graph 430
cost-effectiveness scatterplot 431
incremental cost-effectiveness 3-D mountain 434–437
conversion settings 434–437

incremental cost-effectiveness isocontours 434–437
conversion settings 434
incremental cost-effectiveness scatterplot 432
confidence ellipse, 95% 433
confidence ellipse, variable % 433
converting to 3-D graph 434
converting to isocontour graph 434
options 432
selecting a baseline option 432
selecting a comparator option 432
specifying willingness-to-pay 432
text report 433
incremental net health benefits (NHB), incremental
formulas 438
net health benefits (NHB)
optimality graph 440
probability distributions 439
statistics 439
vs. willingness-to-pay 437–438
probability distributions 430
cost-effectiveness 430
Simulation reports 430–441
simulations using samples and trials
uses other than sensitivity analysis
first-order distributions 511
simulation text report
column descriptions
cost-effectiveness simulations 429–430
simulation using samples 429
simulation using samples and trials 429
simulation using trials 429
simulation using samples 203
details 210
distributions
disabling 214
Sample Some 214
probabilistic sensitivity analysis
using trials vs. cohort/EV calculations 215–218
simulation using trials 203
tracker variables
as inputs (e.g., probabilities) 504
as outputs 504
skeleton tree 128
Skip generation
unskip 123
SMLTREE
bindings 601
importing trees 601
variables 601
spreadsheet 237
startup options 599
status bar 88, 325, 532, 539, 546
calculation method 89
graph windows 83
progress indicator 560
reference 539
string functions 292, 293
sub-models 350
subtree
defined 100
subtrees
collapsing visible 122
expanding hidden 122
pasted vs. cloned copies 282
pasting 110
selecting
for clone master 282
suppressable warnings
reset all 600
suppressed dialogs
re-enabling 532
survival curves 461

T

tab key 141
table
custom distribution 232
Table menu 236
tables
Add Entry dialog 236
create from line graph 83
editing
properties grid 237
entering values 234
Excel module
adding and editing 369
exporting 244
exporting to Excel 367
file format 235
files 243, 518, 598
importing 235
importing data
requirements 236
importing from Excel 367
loading from text 236
lookup method
index-specific 241
interpolation 241
truncation 241
managing files 243–244
moving to another computer 242
options
index off edge is error 241
packaging with tree 242
pasting
problems 237
reporting 244
sharing files 242
storage location 243, 518, 598
tips
column indexes 241
using in formulas 241
Tables Used report 244
terminal node display 563
terminal node numbers 129
tests 310
Text Report command 79
text reports 79–80
exporting to Excel 364
Notes section 79
printing 80
threshold analysis
cost-effectiveness 418
threshold ICER 403, 436
tips
custom interface 589
Markov
tunnel states 486
Markov processes
initial state probabilities 449
reward signs 479
using clones 482
tracker variables 506
tool bar 29, 546
probability wheel 114
reference 538
tornado diagram. *See* sensitivity analysis: tornado diagram
cost-effectiveness 418
incremental 190, 420
net benefits 418
risk percentage 190, 420
text report 190, 420
tracker variables 161, 168, 501–510
analyzing
restrictions 506
as Markov process inputs 504
creating 504–516
using quick menu shortcut 505

defined 504
defining modifications 505
during expected value calculations 506
modifications
order of evaluation 506
order of operations 506, 520
referencing other trackers 506
referencing in the tree 505
transition nodes 455
transition rewards 465
tree. *See* decision tree
explorer view 16
properties view 16, 272, 531
redraw problems. *See* Redraw Window
TreeAge Pro Interactive. *See* ActiveDATALib
compared to Run-time DATA 587
demos 582
technical specifications 582
TreeAgeProLib 384–386, 500. *See also* DATA_Interactive library
ApplicationObj 500
COM library 580
example downloads 384–385
macro examples. *See* TreeAgeProLib
MarkovOutput 500
object documentation. *See* TreeAgeProLib
TreeObj 500
writing Excel macros 356, 362
TreeCalc( ) worksheet function. *See* Excel add-in: TreeCalc( ) worksheet function
tree clipboard 108, 110
switching between 110
viewing 108, 110
TreeObj. *See* TreeAgeProLib: object documentation; *See also* DATA_Interactive library
tree outlines 531
tree properties view 16, 531
tree window 16
Tunnel Crossover binding. *See* Markov state bindings: special: Tunnel Crossover binding
Tunnel Table binding. *See* Markov state bindings: special: Tunnel Table binding
Tunnel Variable binding. *See* Markov state bindings: special: Tunnel Variable binding

U

uncertainty 9
undo 112
last ten actions 112
limitations 112
utility 450
utility function 9

V

value of information 9, 65, 204
cost-effectiveness simulation 441
variables 139–147
basic introduction 138
box
custom interface 594
copying/cutting
definitions 163
correlations 167, 186–189
positive vs. negative 199
removing 167
creating 164
Define Values dialog 164
deleting variables 165
modifying definitions 165
Define Variable window 168
annotating definitions 168
deleting a definition 168
modifying a definition 165
defining 164, 166
at multiple nodes 174
at selected node(s) 175
default numeric value 166
multiple 165
definitions 139
adding/deleting quickly 143–144
displaying below branches 164
deleting 165
displaying in tree 564
edit in Excel 367
Evaluator 158
Excel module
editing and adding 367
formula definitions 139
in influence diagram
defining in mini-tree 337
deterministic nodes 334
hiding 336
nodes 336, 336–342
in sensitivity analysis 148
Insert Variable dialog 160
node-specific definitions 175
numeric definitions 139
properties 166
basic 166
comment 166
name 166
short description 166
show in tree 166
recursive definitions
complex forms 182
how TreeAge evaluates 180
references to deleted variables 165, 166
finding 165, 166
renaming 166
search algorithm
payoffs and rewards 187
probabilities 187
Sliders 159–169
substituting for numeric values 139
tips 174, 186
user-defined functions 288
uses 139
variables (tree)
definitions box
font 567
Variables and Tables dialog 146, 367
Variables Report 157
contents 157
exporting 157
Variables window 169
copying and pasting definitions 163
cutting definitions 163
navigation buttons 162
opening 161
opening a Define Variable window 162
show inherited definitions 162
VBA editor
creating macros 386

W

weights, multi-attribute 547. *See also* decision tree: calculation preferences
willingness-to-pay. *See* cost-effectiveness: willingness-to-pay